

В. Н. Пильщиков



**Сборник
упражнений
по языку
Паскаль**

В. Н. ПИЛЬЩИКОВ

Сборник упражнений по языку Паскаль

*Допущено Государственным комитетом СССР
по народному образованию в качестве учебного пособия
для студентов высших учебных заведений*



МОСКВА «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
1989

ББК 22.18

П32

УДК 519.682(075.8)

Пильщиков В. Н. Сборник упражнений по языку Паскаль: Учеб. пособие для вузов.— М.: Наука. Гл. ред. физ.-мат. лит., 1989.— 160 с.— ISBN 5-02-013995-5

Представлены упражнения и задачи по языку Паскаль и программированию на нем. Рассматриваемая версия Паскаля в целом соответствует международному стандарту этого языка.

Сборник дополняет учебное пособие Абрамова В. Г., Трифонова Н. П., Трифоновой Г. Н. «Введение в язык Паскаль» (М.: Наука, 1988) и составлен с учетом опыта преподавания программирования на факультете вычислительной математики и кибернетики МГУ.

Для студентов младших курсов, специализирующихся в области прикладной математики, и для преподавателей, ведущих практические занятия по программированию.

Табл. 2. Ил. 26. Библиогр. 12 назв.

Рецензенты:

кафедра теоретической кибернетики ЛГУ;

доктор физико-математических наук С. А. Абрамов

П 140400000—107 156-89
053 (02)-89

ISBN 5-02-013995-5

© Издательство «Наука»,
Главная редакция
физико-математической
литературы, 1989

СОДЕРЖАНИЕ

Предисловие	4
1. Числовые типы. Оператор присваивания	7
2. Логический тип	10
3. Простейшие программы	14
4. Операторы: условный, составной, пустой, перехода	18
5. Оператор цикла	22
6. Литерный тип	29
7. Перечисленные и ограниченные типы. Оператор варианта	33
8. Регулярные типы: векторы	39
9. Регулярные типы: матрицы	48
10. Регулярные типы: строки	53
11. Функции и процедуры	58
12. Рекурсия	70
13. Комбинированные типы. Оператор присоединений	74
14. Множественные типы	81
15. Файловые типы	87
16. Ссылочные типы. Списки	96
17. Очереди, стеки, двоичные деревья	109
Ответы и решения	119
Список литературы	155

ПРЕДИСЛОВИЕ

Сборник содержит упражнения и задачи по языку Паскаль [1, 3, 5, 6, 7, 10, 11] и программированию на нем. Набор упражнений составлен с учетом опыта проведения практических занятий по программированию на факультете вычислительной математики и кибернетики Московского университета. Использованы также упражнения из других задачников [2, 4, 8, 12]. Для выполнения большинства упражнений достаточно «здравого смысла», знания элементарной математики и начальных сведений из математического анализа и линейной алгебры. В остальных случаях даются необходимые пояснения.

Сборник является дополнением к учебному пособию [1], поэтому в нем использован тот же порядок рассмотрения тем по Паскалю, что и в этом пособии. Однако сборник может быть использован и независимо от пособия; при этом следует учитывать, что рассматриваемая версия Паскаля соответствует международному стандарту этого языка (ISO 7185—1983, уровень 0) со следующими изменениями: исключены записи с вариантами, в алфавит языка введены русские буквы.

Упражнения сборника охватывают все темы по Паскалю и сгруппированы в 17 разделов. Каждый раздел сборника посвящен одному из понятий языка (или нескольким тесно связанным понятиям). Раздел начинается с упражнений, акцентирующих внимание на синтаксисе и семантике этого понятия, на его назначении и типичных случаях использования, на ошибках, встречающихся у начинающих программистов. Здесь же рассматриваются

приемы программирования и алгоритмы, часто встречающиеся на практике. Для многих из этих упражнений в конце сборника приведены ответы или решения; такие упражнения помечены звездочкой. В конце раздела собраны упражнения на составление программ. Они предназначены для самостоятельной работы, поэтому их решения не приводятся, однако в необходимых случаях даются рекомендации или подсказки.

Нумерация упражнений в каждом разделе начинается с 1. Многие упражнения имеют варианты, которые обозначаются начальными буквами русского алфавита. При ссылке на упражнение указывается номер раздела и номер упражнения в разделе, а если надо, то и буква варианта.

Для сокращения формулировок упражнений в сборнике используются следующие соглашения.

Фраза «найти ошибки в программе» означает, что надо не только указать все ошибки в приведенной программе, но и объяснить, какие при этом правила Паскаля нарушины. Следует учитывать, что ошибки могут быть не только синтаксическими или семантическими, но и ошибками периода счета (например, в программе используется переменная, которой не присвоено начальное значение).

Если формулировка упражнения начинается с предложения «Программа.», то решение предложением задачи должно быть описано в виде подной программы — с вводом-выводом, с описанием всех используемых типов, переменных и т. п. Встречающиеся в таких упражнениях фразы типа «дано число n » или «для заданных массивов x и y » указывают на то, что это число или эти массивы являются исходными данными программы и должны быть в ней введены. Результатами же являются величины, которые надо «вычислить», «определить», «построить» и т. п. Формат печати результатов, как правило, не фиксируется и должен определяться автором программы. В некоторых упражнениях решение задачи требуется описать в виде процедуры или функции, это всегда оговаривается явно.

Исходные данные и результаты задачи в таком случае должны быть объявлены параметрами процедуры или функции. В начале этих упражнений приводится описание всех используемых в задаче констант и нестандартных типов. В остальных упражнениях решение задачи должно быть оформлено как фрагмент программы — как последовательность операторов без описаний и ввода-вывода, если они не требуются явно. В начале таких упражнений обычно дается описание переменных, обозначающих исходные данные и результаты задачи, и описание используемых констант и нестандартных типов.

В приводимых в сборнике программах и их фрагментах все идентификаторы набраны прямым шрифтом, в тексте же они выделены курсивом.

Автор выражает благодарность сотрудникам факультета ВМК МГУ В. Г. Абрамову, Е. В. Зиме, В. В. Игнатову, В. И. Родину, Т. В. Руденко и М. Ю. Семенову за полезные советы и помощь при подготовке сборника.

I. ЧИСЛОВЫЕ ТИПЫ. ОПЕРАТОР ПРИСВАИВАНИЯ

1.1*. Записать на Паскале следующие числа (в дробной части указывать до 4 цифр):

- а) 5!; б) LXIV; в) 6,38;
- г) -0,7(4); д) 11/4; е) -1/6;
- ж) $\sqrt{2}$; з) π ; и) $5 \cdot 10^6$;
- к) $-24,8 \cdot 10^{-7}$; л) 10^4 ; м) $1/100000$.

1.2*. Записать следующие числа без десятичного порядка:

- а) -0.00027E+4; б) 666E-3; в) 1E1

1.3*. Указать неправильные записи чисел:

- а) 0006; б) -0; в) 7,0; г) 7.;
- д) +0.3; е) .3; ж) 2/3; з) E-1;
- и) 8E0; к) 0E-4; л) 2*E5; м) e

1.4. Есть ли разница (с точки зрения языка Паскаль) между числами 100 и 100.0, между 20 и 2E1? По какому признаку (величине или форме записи) целые числа отличаются от вещественных?

1.5. Привести примеры положительного и отрицательного целых чисел, не представимых в Паскале.

1.6*. Можно ли утверждать, что в Паскале значение выражения $(1/3)*3-1$ равно нулю?

1.7. Почему при записи формул на Паскале их «вытягивают» в линию? Почему знак умножения всегда выписывают явно (например, пишут $a*t$, а не at)?

1.8*. Записать на Паскале следующие формулы:

- а) $a+bx+cyz$; б) $[(ax-b)x+c]x-d$;

в) $\frac{ab}{c} + \frac{c}{ab}$; г) $\frac{x+y}{a_1} \cdot \frac{a_2}{x-y}$;

д) $10^4\alpha - 3\frac{1}{5}\beta$; е) $\left(1 + \frac{x}{2!} + \frac{y}{3!}\right) / \left(1 + \frac{2}{3+xy}\right)$.

1.9*. Записать в общепринятой форме:

- а) $(p+q)/(r+s)-p*q/(r*s)$;
- б) $1E3+\text{beta}/(x^2-\text{gamma}*\text{delta})$

1.10*. Вычислить: $24/(3*4) - 24/3/4 + 24/3*4$

1.11. Почему в Паскале факториал от 10 нельзя записать в виде $10!$ или $1*2*3*...*10!$?

1.12*. Сколько операций выполняется при вычислении выражения

$$(x+1/2)*(y+7/10)-3/4?$$

Как сократить число операций?

1.13. Почему в Паскале аргумент функции всегда записывают в скобках (например, пишут $\ln(5)$, а не $\ln 5$)?

1.14. Записать на Паскале следующие формулы:

а) $(1+x)^2$; б) $\sqrt{1+x^2}$; в) $|a+bx|$;

г) $\sin 8$; д) $\cos^2 x^2$; е) $\operatorname{tg} x$;

ж) $\log_5 \frac{x}{5}$; з) $\operatorname{ch} x$; и) $\operatorname{arcctg} 10^3$;

к) $\arcsin x$.

1.15. Записать на Паскале следующие формулы ($x > 0$):

а) x^{-1} ; б) x^4 ; в) x^{-2} ; г) x^5 ;

д) x^{100} ; е) 2^{1+x} ; ж) $x^{\sqrt{2}}$; з) $\sqrt[3]{1+x}$.

1.16*. Как записать на Паскале величину основания натуральных логарифмов (число e), если Вы забыли цифры этого числа? А как записать число π в подобной ситуации?

1.17*. Как на Паскале записать синус от x градусов?

1.18. Записать на Паскале следующие формулы:

а) $\sqrt[8]{x^6+8x}$; б) $\frac{xyz-3,3|x+\sqrt[4]{y}|}{10^7+\sqrt{\lg 4!}}$;

в) $\frac{\beta+\sin^2 \pi^4}{\cos 2+\operatorname{ctg} \gamma}$.

1.19. Записать в общепринятой форме:

а) $(-b+\operatorname{sqrt}(\operatorname{sqrt}(b)-4*a*c))/(2*a)$;

б) $a/b*(c+d)-(a-b)/b/c+1E-8$;

в) $x1+\operatorname{arctan}(y2-\alpha)/2*\operatorname{abs}(x4-\ln(5)*y5)/\operatorname{exp}(-1)$

1.20. Записать на Паскале соответствующие операторы присваивания:

а) $y=1+x+\frac{x^3}{2!}+\frac{x^5}{3!}+\frac{x^4}{4!}$;

б) $f=6,673 \cdot 10^{-6} \cdot \frac{m_1 \cdot m_2}{r^2}$;

в) $b=e^{|x-y|}+\ln(1+e)\log_2 \operatorname{tg} 2$.

1.21. Записать операторы присваивания, которые переменной d присваивают:

а) среднее арифметическое чисел x , y , z ;

б) расстояние между точками с координатами (x_1, y_1) и (x_2, y_2) ;

в) корень уравнения $\arctg(1+\ln x)=\sqrt{2}$;

г)* площадь треугольника со сторонами a , b и c .

1.22. Какое значение будет иметь переменная x после выполнения операторов $x:=10$; $x:=x+3$?

1.23. Записать оператор присваивания, который меняет знак у значения переменной t .

1.24. Чему равны значения переменных x и y после выполнения операторов $x:=2$; $y:=5$; $x:=y$; $y:=x$?

1.25*. Поменять местами значения переменных x и y .

1.26. Поменять местами значения переменных x , y и z так, чтобы в x оказалось значение переменной y , в y — значение переменной z , а в z — прежнее значение переменной x .

1.27*. Вычислить значения выражений:

а) $\text{trunc}(6.9)$; б) $\text{round}(6.9)$;

в) $\text{trunc}(6.2)$; г) $\text{round}(6.2)$;

д) $\text{trunc}(-1.8)$; е) $\text{round}(-1.8)$;

ж) $\text{round}(0.5)$; з) $\text{round}(-0.5)$

1.28*. Переменной d присвоить дробную часть положительного числа x .

1.29*. Вычислить значения выражений:

а) $20 \text{ div } 6$; б) $20 \text{ mod } 6$;

в) $20 \text{ div } 4$; г) $20 \text{ mod } 4$;

д) $2 \text{ div } 5$; е) $2 \text{ mod } 5$;

ж) $123 \text{ div } 0$; з) $3.0 \text{ mod } 3$

1.30. Определить операцию div через другие операции и стандартные функции.

1.31*. Указать порядок выполнения операций в выражении

$-a \text{ mod } b + a \text{ div } b * c$

1.32. Вычислить значения выражений:

а) $3*7 \text{ div } 2 \text{ mod } 7/3 - \text{trunc}(\sin(1))$;

б)* $\text{succ}(\text{round}(5/2)) - \text{pred}(3))$

1.33*. Определить тип (целый или вещественный) выражения:

а) $1+0.0$; б) $20/4$; в) $\text{sqr}(4)$;

г) $\text{sqr}(5.0)$; д) $\text{sqrt}(16)$; е) $\sin(0)$;

ж) $\text{succ}(-2)$; з) $\text{trunc}(-3.14)$

1.34. В языке Паскаль значением вещественной переменной (скажем, x) может быть только вещественное число и в то же время допускается оператор присваивания, который вещественной переменной присваивает целое число (например, $x:=7$). Как в языке устраняется это противоречие?

1.35*. Если y —вещественная переменная, а n —целая, то какие из следующих операторов присваивания правильные, а какие нет и почему?

- а) $y:=n+1$; б) $n:=y-1$; в) $n:=4.0$;
- г) $y:=\text{trunc}(y)$; д) $n:=n \text{ div } 2$;
- е) $y:=y \text{ div } 2$; ж) $n:=n/2$; з) $n:=\text{sqr}(\sqrt{n})$

1.36. Правильны ли следующие операторы присваивания? Ответ обосновать.

- а)* $k:=k \bmod 3+k*\cos(0)$;
- б) $x:=x*2 \text{ div } 6+x/4$

1.37*. Присвоить целой переменной h третью от конца цифру в записи положительного целого числа k (например, если $k=130985$, то $h=9$).

1.38. Присвоить целой переменной d первую цифру из дробной части положительного вещественного числа x (так, если $x=32.597$, то $d=5$).

1.39. Целой переменной s присвоить сумму цифр трехзначного целого числа k .

1.40. Идет k -я секунда суток. Определить, сколько полных часов (h) и полных минут (m) прошло к этому моменту (например, $h=3$ и $m=40$, если $k=13257=3*3600+40*60+57$).

1.41. Определить f —угол (в градусах) между положением часовой стрелки в начале суток и ее положением в h часов, m минут и s секунд ($0 \leq h \leq 11$, $0 \leq m, s \leq 59$).

1.42. Определить h —полное количество часов и m —полное количество минут, прошедших от начала суток до того момента (в первой половине дня), когда часовая стрелка повернулась на f градусов ($0 \leq f < 360$, f —вещественное число).

1.43. Пусть k —целое от 1 до 365. Присвоить целой переменной n значение 1, 2, ..., 6 или 7 в зависимости от того, на какой день недели (понедельник, вторник, ..., субботу или воскресенье) приходится k -й день невисокосного года, в котором 1 января—понедельник.

1.44. Поменять местами значения целых переменных x и y , не используя дополнительные переменные.

2. ЛОГИЧЕСКИЙ ТИП

2.1*. Вычислить значения выражений:

- а) $\text{sqr}(x)+\text{sqr}(y) \leq 4$ при $x=0.3$, $y=-1.6$;
- б) $k \bmod 7 = k \text{ div } 5 - 1$ при $k = 15$;
- в) $\text{odd}(\text{trunc}(10*p))$ при $p = 0.182$

2.2*. Записать на Паскале отношение, истинное при выполнении указанного условия и ложное в противном случае:

- а) целое k делится на 7;
 - б) уравнение $ax^2+bx+c=0$ ($a \neq 0$) не имеет вещественных корней;
 - в) точка (x, y) лежит вне круга радиуса r с центром в точке $(1, 0)$;
 - г) натуральное n является полным квадратом.
- 2.3*. Вычислить значения выражений:

- а) $\text{not odd}(n)$ при $n=0$;
- б) $t \text{ and } (p \bmod 3 = 0)$ при $t=\text{true}$, $p=101010$;
- в) $(x*y < 0) \text{ and } (y > x)$ при $x=2$, $y=1$;
- г) $(x*y < 0) \text{ or } (y > x)$ при $x=2$, $y=1$;
- д) $a \text{ or } (\text{not } b)$ при $a=\text{false}$, $b=\text{true}$.

2.4*. Записать на Паскале выражение, истинное при выполнении указанного условия и ложное в противном случае:

- а) $0 < x < 1$;
- б) $x = \max(x, y, z)$;
- в) $x \neq \max(x, y, z)$ (операцию not не использовать);
- г) хотя бы одна из логических переменных a и b имеет значение true ;
- д) обе логические переменные a и b имеют значение true .

2.5. Доказать тождества:

- а)* $a \text{ and } (\text{not } a) = \text{false}$;
- б) $a \text{ or } (\text{not } a) = \text{true}$;
- в) $\text{not } (\text{not } a) = a$;
- г) $\text{true or } a = \text{true}$;
- д) $\text{false and } a = \text{false}$;
- е) $a \text{ or } a = a$.

2.6*. Вычислить:

- а) $\text{true or } (1/0 > 0)$;
- б) $(1/0 > 0) \text{ or } \text{true}$

2.7*. Объяснить ошибки в следующих записях:

- а) $1 \text{ and } 0$;
- б) $\text{true} + \text{false}$;
- в) $\text{true} < 0$;
- г) $\text{not } 2 = 5$;
- д) $x > 0 \text{ or } y = 4$;
- е) $\text{not not } b \text{ or or } d$

2.8. Указать порядок выполнения операций при вычислении выражения:

- а)* $a \text{ and } b \text{ or not } c \text{ and } d$;
- б) $(x >= 0) \text{ or } t \text{ and } \text{odd}(x) \text{ or } (y * y < > 4)$

2.9. Вычислить следующие выражения при $a=\text{true}$ и $b=\text{false}$:

- а) $a \text{ or } b \text{ and not } a$;
- б) $(a \text{ or } b) \text{ and not } a$;
- в) $\text{not } a \text{ and } b$;
- г) $\text{not } (\text{a and b})$

2.10. Записать на Паскале выражение, истинное при выполнении указанного условия и ложное в противном случае:

- а) x принадлежит отрезку $[0, 1]$;
- б) x лежит вне отрезка $[0, 1]$;
- в)* x принадлежит отрезку $[2, 5]$ или $[-1, 1]$;
- г)* x лежит вне отрезков $[2, 5]$ и $[-1, 1]$;
- д) каждое из чисел x, y, z положительно;
- е) хотя бы одно из чисел x, y и z положительно;
- ж) ни одно из чисел x, y и z не является положительным;
- з) только одно из чисел x, y и z положительно;
- и) логическая переменная a имеет значение *true*, а логическая переменная b имеет значение *false*;
- к)* год с порядковым номером y является високосным (год високосный, если его номер кратен 4, однаково из кратных 100 високосными являются лишь кратные 400; например, 1700, 1800 и 1900—невисокосные годы, а 2000—високосный).

2.11. Нарисовать на плоскости (x, y) область, в которой и только в которой истинно указанное выражение:

- а)* $(y >= x) \text{ and } (y + x >= 0) \text{ and } (y <= 1)$;
- б) $(\text{sqr}(x) + \text{sqr}(y) < 1)$ or
 $(y > 0) \text{ and } (\text{abs}(x) <= 1)$;
- в) $(\text{trunc}(y) = 0) \text{ and } (\text{round}(x) = 0)$

2.12*. Записать на Паскале выражение, зависящее от x и y , которое принимает значение *true*, когда точка с координатами x и y принадлежит заштрихованной области (см. рис. 1).

2.13*. Вычислить значения выражений:

- а) $\text{false} < \text{true}$;
- б) $\text{ord}(\text{false}) = 1$;
- в) $\text{pred}(\text{true})$;
- г) $\text{ord}(\text{succ}(\text{false})) > 0$

2.14*. Вычислить значения выражений:

- а) $\text{not}(\text{pred}(c) \text{ or } (\text{ord}(c) = 1))$ при $c = \text{true}$;
- б) $(p < \text{true}) = (q = \text{false})$ при $p = q = \text{true}$,
- в) $a \text{ and } b > a \text{ or } b$ при $a = \text{false}, b = \text{true}$.

2.15. Записать на Паскале выражение, истинное при выполнении указанного условия и ложное иначе:

- а)* целые n и k имеют одинаковую четность;
- б)* только одна из логических переменных a и b имеет значение *true*;
- в) только одна из логических переменных a, b и c имеет значение *true*.

2.16. Нарисовать на плоскости (x, y) область, в которой и только в которой истинно указанное выражение:

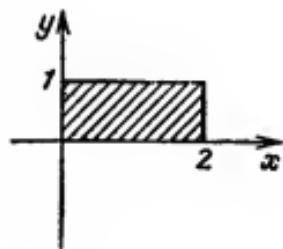
- a)* $(\text{abs}(x) \leq 1) > (\text{abs}(y) \geq 1)$;
 б) $(\text{sqr}(x) + \text{sqr}(y) \leq 4) = (y \leq -x)$

2.17. Доказать тождества:

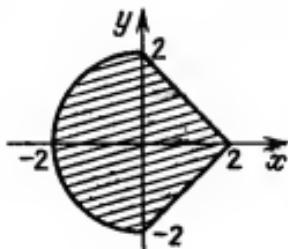
- а) $\text{not}(a \text{ or } b) = (\text{not } a) \text{ and } (\text{not } b)$;
 б) $a \text{ and } (b \text{ or } c) = (a \text{ and } b) \text{ or } (a \text{ and } c)$;
 в)* $a \leq b = \text{not } a \text{ or } b$;
 г) $a \text{ and } b = (a < \text{true}) < b$;
 д) $\text{not } a = a < \text{true}$.

2.18. Преобразовать указанное выражение к виду, не содержащему знаков отношения (a и b — логические переменные):

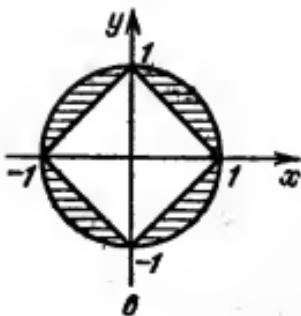
- а)* $a < b$; б) $a = b$; в) $(a < b) = a$



а



б



в

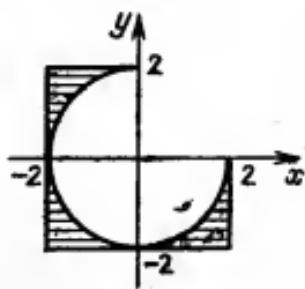


Рис. 1

2.19. Если $a = \text{true}$ и $x = 1$, то какое значение получит логическая переменная d после выполнения оператора присваивания?

- а)* $d := x < 2$; б) $d := \text{not } a \text{ or } \text{odd}(x)$;
 в) $d := \text{ord}(a) <> x$

2.20. Написать оператор присваивания, в результате выполнения которого логическая переменная t получает значение true , если выполняется указанное условие, и значение false иначе:

- а) числа x , y , z равны между собой;

- б) из чисел x , y , z только два равны между собой;
 в) x — положительное число;
 г) p делится нацело на q (p и q — натуральные числа);
 д) уравнение $ax^2+bx+c=0$, где a , b и c могут равняться 0, имеет ровно один корень;
 е) цифра 5 входит в десятичную запись трехзначного целого числа k ;
 ж) поля $(g1, g1)$ и $(g2, g2)$ шахматной доски имеют одинаковый цвет ($g1$, $g1$, $g2$ и $g2$ — целые от 1 до 8);
 з) ферзь, расположенный на поле $(g1, g1)$ шахматной доски, «бьет» поле $(g2, g2)$.

3. ПРОСТЕЙШИЕ ПРОГРАММЫ

3.1*. Что будет напечатано программой

```
program корни (input, output);
var b, c, d: real;
begin read(b, c);
d:=sqrt(sqrt(b)-4*c);
writeln('x1=', (-b+d)/2, ' x2=', (-b-d)/2)
end.
```

если в качестве исходных данных заданы числа 1.0 и -2.0?

3.2*. Написать программу, которая вводит два вещественных числа, вычисляет и печатает коэффициенты приведенного квадратного уравнения, корнями которого являются эти числа.

3.3. Что будет напечатано программой

```
program less (input, output);
var x:real; t:boolean;
begin read(x); t:=x<round(x);
read(x); t:=t and (x<trunc(x));
writeln(t)
end.
```

если для ввода заданы числа 1.5 и -0.8?

3.4. Написать программу, которая печатает *true* или *false* в зависимости от того, имеют три заданных целых числа одинаковую четность или нет.

3.5*. Можно ли для программы

```
program умножение (input, output);
var a:integer; b:real;
begin read(a, b); writeln(a*b) end.
```

задать в качестве исходных данных числа 5.0 и 6.2?

3.6. Ответить на следующие вопросы:

а) почему параметрами процедуры ввода *read* могут быть только переменные, а не числа или выражения, скажем, вида $x+1$?

б) в процедуре ввода указывается имя переменной, которой надо присвоить введенную величину, но не указывается, какую именно из заданных во входном файле величин надо ввести; как же становится известным, какую величину надо вводить?

в) если в качестве исходных данных задано пять чисел, то можно ли ввести пятое из них, не вводя первые четыре?

3.7*. Что будет напечатано программой

```
program aba (input, output);
  var a, b:integer;
begin read(a, b, a); writeln(a, b, a) end.
```

если для ввода заданы числа 1, 2 и 3?

3.8*. Имеется программа

```
program печать (input, output);
  var x:integer;
begin x:=2; writeln('x+1') end.
```

Что она напечатает: 3 или $x+1$?

3.9. Когда процедура вывода *writeln*(x_1, \dots, x_n) осуществляет перевод строки: до печати первого параметра x_1 или после печати последнего параметра x_n ?

Нарисовать расположение чисел, напечатанных следующей последовательностью процедур:

```
write(1); write(2, 3); writeln(4); write(5, 6);
writeln; writeln(7, 8)
```

3.10*. Написать программу, которая для заданного целого числа a печатает следующую таблицу

a		
a^3	a^6	
a^6	a^3	a

3.11*. Программа

```
program степени (output);
  const e=2.71828;
  var e2:real;
begin e2:=e*e; writeln(e,e2,e*e2,e2*e2) end.
```

печатает первые четыре степени числа e . Какие изменения (по возможности минимальные) надо внести в программу, чтобы она печатала первые четыре степени числа π ?

3.12*. Написать программу, которая печатает *true* или *false* в зависимости от того, больше число e^x числа π^x или нет. (Числа e и π с точностью 10^{-6} описать как константы.)

3.13. Написать программу, которая вычисляет периметр и площадь правильного 17-угольника, вписанного в окружность заданного радиуса.

Что надо изменить в программе для того, чтобы она правильно решала эту же задачу для 25-угольника?

3.14. Ответить на следующие вопросы.

а) Верно ли, что в паскаль-программе надо описывать все используемые в ней переменные?

б) Какую информацию извлекает транслятор из описания переменных и как он ее использует?

в) Какие значения имеют переменные в начале выполнения программы?

г) Можно ли менять значения констант?

д) В описании констант их типы не указываются. Как же определяются типы констант?

е) Почему не надо описывать константы *maxint*, *true*, *false*?

3.15*. Найти ошибки в каждой из следующих программ.

а) program A (output);
const d=5;
begin d:=sqr(d);
writeln('d**2=',d) end.

б) program Б (input, output);
const k=true; var x:real;
begin read(x); writeln(ord(x)=k) end.

в) program В (input, output);
var a,b,c:integer;
begin read(a,b); writeln((a+b+c)/3) end.

г) program Г (input, output);
var x:real;
begin read(x); y:=sqrt(x)+1;
writeln(y) end.

д) program Д (input, output);
const B=2.5; var a,b,c:real;
begin read(a,c); writeln(a*c>b) end.

3.16. Найти ошибки в следующей программе:

```
program ошибки (input, output)  
const π = 3.14159;  
var a,b:integer;
```

```
begin read(A); d:=odd(l*0) and b>a;  
writeln(d) end.
```

3.17. Что будет напечатано следующей программой, если для ввода было задано число 31.7?

```
program time(input,output);var  
fi:real;h,m:integer;begin read  
(fi); h:=trunc(fi/30);m:=trunc(  
(fi-30*h)/0.5);writeln(h,' ',m  
)end.
```

Записать текст этой программы в более наглядной форме.

3.18. Написать программы для решения следующих задач:

а) вычислить значение производной функции x^x в заданной точке a ($a > 0$);

б) для заданного a вычислить принадлежащий интервалу $(\pi, 2\pi)$ корень уравнения

$$\ln(\operatorname{ctg} x - 1) = a;$$

в) вычислить дробную часть среднего геометрического трех заданных положительных чисел;

г) по заданным коэффициентам и правым частям уравнений системы

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

найти ее решение в предположении, что определитель системы не равен нулю;

д) вычислить длину окружности, площадь круга и объем шара одного и того же заданного радиуса;

е) вычислить периметр и площадь прямоугольного треугольника по длинам двух катетов;

ж) по координатам трех вершин некоторого треугольника найти его площадь и периметр;

з) по длинам двух сторон некоторого треугольника и углу (в градусах) между ними найти длину третьей стороны и площадь этого треугольника;

и) найти произведение цифр заданного четырехзначного числа;

к) определить число, полученное выписыванием в обратном порядке цифр заданного трехзначного числа.

3.19. Для решения следующих задач написать программы, которые печатают *true* или *false* в зависимости от того, выполняются или нет указанные условия!

- а) для произвольных вещественных чисел a , b и c определить, имеет ли уравнение $ax^2 + bx + c = 0$ хотя бы одно вещественное решение;
 б) для заданных чисел p , a и b ($a < b$) определить, имеет ли уравнение $\operatorname{arctg}(2x - |p|) = \sqrt{2}$ корень на отрезке $[a, b]$;
 в) определить, равна ли сумма двух первых цифр заданного четырехзначного числа сумме двух его последних цифр;
 г) определить, равен ли квадрат заданного трехзначного числа кубу суммы цифр этого числа;
 д) определить, есть ли среди первых трех цифр из дробной части заданного положительного вещественного числа цифра 0;
 е) определить, есть ли среди цифр заданного трехзначного числа одинаковые;
 ж) даны три произвольных числа. Определить, можно ли построить треугольник с такими длинами сторон;
- з) даны координаты (как целые от 1 до 8) двух полей шахматной доски. Определить, может ли конь за один ход перейти с одного из этих полей на другое.

4. ОПЕРАТОРЫ: УСЛОВНЫЙ, СОСТАВНОЙ, ПУСТОЙ, ПЕРЕХОДА

4.1. Записать указанное действие в виде одного условного оператора:

$$a)* y = \begin{cases} \cos^2 x & \text{при } 0 < x < 2, \\ 1 - \sin x^2 & \text{иначе;} \end{cases}$$

б)* переменной x присвоить корень уравнения
 $\arcsin(1 + \ln x) = a$,

если такой существует;

в) перераспределить значения переменных x и y так, чтобы в x оказалось большее из этих значений, а в y — меньшее;

$$\gamma)* d = \max(a, b, c);$$

$$\delta)* z = \begin{cases} \max(x, y) & \text{при } x < 0, \\ \min(x, y) & \text{при } x \geq 0; \end{cases}$$

е) переменной k присвоить номер четверти плоскости, в которой находится точка с координатами x и y ($xy \neq 0$);

ж) известно, что из четырех чисел a_1 , a_2 , a_3 и a_4 одно отлично от трех других, равных между собой; присвоить номер этого числа переменной n .

4.2. Вычисление $y=f(x)$, где функция $f(x)$ задана графиком (рис. 2), описать в виде одного оператора.

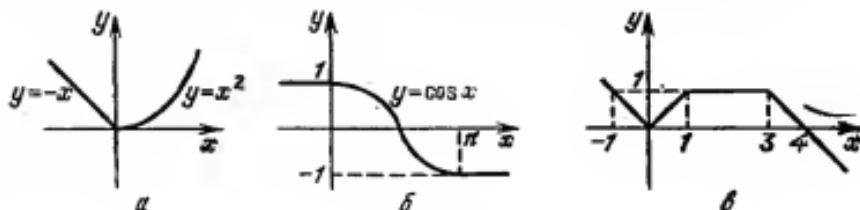


Рис. 2

4.3. Записать последовательность операторов для решения указанной задачи:

а)* по номеру y ($y>0$) некоторого года определить c — номер его столетия (учесть, что, к примеру, началом XX столетия был 1901, а не 1900 год);

$$\text{б) } u = \frac{\max^2(x,y,z) - 2x \cdot \min(x,y,z)}{\sin 2 + \max(x,y,z)/\min(x,y,z)};$$

в)* если уравнение $ax^2+bx+c=0$ ($a \neq 0$) имеет вещественные корни, то логической переменной t присвоить значение *true*, а переменным $x1$ и $x2$ — сами корни, иначе же переменной t присвоить *false*, а значения переменных $x1$ и $x2$ не менять;

г) считая, что стандартные функции \sin и \cos применимы только к аргументам из отрезка $[0, \pi/2]$, вычислить $y=\sin x$ для произвольного числа x ;

д)* значения переменных a , b и c поменять местами так, чтобы оказалось $a \geq b \geq c$.

4.4*. Какое значение будет иметь переменная z после выполнения операторов

```
z:=0;
if x>0 then if y>0 then z:=1 else z:=2
```

при следующих значениях переменных x и y :

а) $x=y=1$; б) $x=1$, $y=-1$; в) $x=-1$, $y=1$.

4.5*. Если значение переменной w не равно 0 и при этом котангенс от w меньше 0.5, тогда поменять знак у w , а если значение w равно 0, тогда присвоить w значение 1.

4.6. Указать ошибки!

- if $1 < x < 2$ then $x := x + 1$; $y := 0$;
else $x := 0$; $y := y + 1$;
- if $1 < x$ and $x < 2$
then begin $x := x + 1$; $y := 0$ end;
else begin $x := 0$; $y := y + 1$ end

4.7*. Логической переменной b присвоить значение *true*, если числа x и y равны, и значение *false* иначе.

4.8*. Записать условный оператор, который эквивалентен оператору присваивания

$x := a \text{ or } b \text{ and } c$

(все переменные — логические) и в котором не используются логические операции (например, оператору $x := \text{not } a$ эквивалентен оператор $\text{if } a \text{ then } x := \text{false} \text{ else } x := \text{true}$).

4.9. Записать оператор присваивания, эквивалентный условному оператору

$\text{if } a \text{ then } x := b \text{ else } x := c$

где все переменные — логического типа.

4.10. Написать программу для решения указанной задачи:

а) для заданного числа a найти корень уравнения $f(x) = 0$, где

$$f(x) = \begin{cases} 2ax + |a - 1| & \text{при } a > 0, \\ \frac{e^x}{\sqrt{1+a^2}} - 1 & \text{иначе;} \end{cases}$$

б) дано число x . Напечатать в порядке возрастания числа $\sin x$, $1 + |x|$ и $(1+x^2)^x$;

в) даны числа $a_1, b_1, c_1, a_2, b_2, c_2$. Напечатать координаты точки пересечения прямых, описываемых уравнениями $a_1x + b_1y = c_1$ и $a_2x + b_2y = c_2$, либо сообщить, что эти прямые совпадают, не пересекаются или вовсе не существуют;

г) даны числа a, b и c ($a \neq 0$). Найти вещественные корни уравнения $ax^2 + bx + c = 0$. Если корней нет, то сообщить об этом;

д) даны произвольные числа a, b и c . Если нельзя построить треугольник с такими длинами сторон, то напечатать 0, иначе напечатать 3, 2 или 1 в зависимости от того, равносторонний это треугольник, равнобедренный или какой-либо иной;

е) дано целое k от 1 до 180. Определить, какая цифра находится в k -й позиции последовательности

10111213...9899,

в которой выписаны подряд все двухзначные числа;

ж) дано натуральное k . Определить k -ю цифру в последовательности

110100100010000100000...,

в которой выписаны подряд степени 10.

4.11*. Есть ли в следующих текстах пустые операторы?

- а) if $x > 0$ then $x := 2$ else; $y := x + 1$;
- б) if odd(k) then else $k := 0$;
- в) begin $x := 2$; $y := 5$; end;
- г) begin $a := \text{true}$; ; $b := b$ or a end;
- д) begin if $x = 0$ then goto 1; $y := x$; 1: end

4.12. Допустимы ли в Паскале следующие составные операторы?

- а) begin end;
- б) begin $x := 0$ end;
- в) begin ; end

4.13*. Если $n = 3$, то какое значение будет иметь переменная f после выполнения следующего составного оператора?

```
begin f:=1; i:=2;
      1: if i>n then goto 9;
          f:=f*i; i:=i+1; goto 1;
      9: end
```

4.14. Выписать последовательность операторов для решения каждой из следующих задач:

а)* вычислить c — наибольший общий делитель натуральных чисел a и b .

б) найти u — первый отрицательный член последовательности $\cos(\operatorname{ctg} n)$, $n = 1, 2, 3, \dots$;

в)* вычислить $p = (1 - 1/2^2)(1 - 1/3^2) \dots (1 - 1/n^2)$, $n > 2$;

г) вычислить $y = \cos(1 + \cos(2 + \dots + \cos(39 + \cos 40) \dots))$.

4.15. Ответить на следующие вопросы.

а) Верно ли, что в качестве метки можно использовать любое положительное целое число?

б) Обязательно ли описывать все метки, которыми помечены операторы программы?

в) Верно ли, что в разделе меток все описываемые метки должны быть перечислены по возрастанию?

4.16. Найти ошибки в программе:

```
program errors (input,output);
  const L=18;
  label 18,8;
  var x,y:integer;
begin y:=0;
  8: read(x); if x<0 then goto L;
    y:=y+x; goto 18;
  L: writeln(y)
end.
```

4.17. Найти ошибки в следующих операторах:

- a) begin 77: if $|x|<1$ then goto 5
begin x:=x/2; goto 77; 5: y:=sin(x) end
end;
- b) 34: if $0 < x < 2$
then if $y < 1$ then goto 34 else goto 15
else 15: y:=sqr(y)

4.18*. Программа. Дано 50 вещественных чисел. Найти величину наибольшего из них.

4.19. Программа. Дано целое $n > 0$, за которым следует n вещественных чисел. Определить, сколько среди них отрицательных.

4.20. Программа. Даны непустая последовательность положительных целых чисел, за которой следует 0 (это признак конца последовательности). Вычислить среднее геометрическое этих чисел.

5. ОПЕРАТОР ЦИКЛА

5.1*. Вычисление $f = 10!$ описать каждым из трех вариантов оператора цикла.

5.2*. Определить значение переменной s после выполнения следующих операторов:

- a) $s := 0; i := 0;$
while $i < 5$ do $i := i + 1; s := s + 1/i;$
- b) $s := 0; i := 1;$
while $i > 1$ do begin $s := s + 1/i; i := i - 1$ end;
- v) $s := 0; i := 1;$
repeat $s := s + 1/i; i := i - 1$ until $i \leq 1;$
- r) $s := 1; n := 1;$
for $i := 2$ to n do $s := s + 1/i$

5.3*. Выписать фрагмент программы для решения указанной ниже задачи и обосновать, почему был выбран тот или иной вариант оператора цикла:

а) вычислить c —наибольший общий делитель натуральных чисел a и b ;

б) найти u —первый отрицательный член последовательности $\cos(\operatorname{ctg} n)$, $n=1, 2, 3 \dots$;

в) вычислить $p=(1-1/2^2)(1-1/3^2) \dots (1-1/n^2)$, $n>2$;

г) вычислить $y=\cos(1+\cos(2+\dots+\cos(39+\cos 40)\dots))$.

5.4*. Имеется целое $k \geq 0$. Вычислить x — k -й член последовательности $\{x_n\}$, где $x_0=1$, $x_n=nx_{n-1}+1/n$ при $n=1, 2, 3, \dots$

5.5*. С точностью 10^{-6} вычислить x —наименьший положительный корень уравнения $\operatorname{tg} x=x$, используя метод деления отрезка пополам.

5.6. Подсчитать k —количество цифр в десятичной записи целого неотрицательного числа n .

5.7. Логической переменной t присвоить значение *true* или *false* в зависимости от того, является ли натуральное число k степенем 3 или нет.

5.8. Программа. Даю 100 вещественных чисел. Вычислить разность между максимальным и минимальным из них.

5.9*. Программа. Даю непустая последовательность различных натуральных чисел, за которой следует 0. Определить порядковый номер наименьшего из них.

5.10. Программа. Даю целое $n > 0$ и последовательность из n вещественных чисел, среди которых есть хотя бы одно отрицательное число. Найти величину наибольшего среди отрицательных чисел этой последовательности.

5.11*. Вычислить по схеме Горнера:

а) $y=x^{10}+2x^9+3x^8+\dots+10x+11$;

б) $y=11x^{10}+10x^9+9x^8+\dots+2x+1$.

5.12. Программа. Даю натуральное число n и вещественные числа t , a_0 , a_1 , ..., a_n . Вычислить значение многочлена

$$a_0x^n+a_1x^{n-1}+\dots+a_{n-1}x+a_n$$

и его производной в точке $t^2+0.5$.

5.13. Вычислить:

а) $y=(2n-1)!!=1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-1)$, $n > 0$;

б) $y=(2n)!!=2 \cdot 4 \cdot \dots \cdot (2n)$, $n > 0$;

в) $y=n!!$, $n > 0$.

5.14. Вычислить:

$$y = \sqrt{3 + \sqrt{6 + \dots + \sqrt{96 + \sqrt{99}}}}$$

5.15. Вычислить:

a)* $y = \cos x + \cos x^3 + \cos x^9 + \dots + \cos x^{3^n}$;

б)* $y = 1! + 2! + 3! + \dots + n!$ ($n > 1$);

в) y — первое из чисел $\sin x, \sin \sin x, \sin \sin \sin x, \dots$, меньшее по модулю 10^{-4} .

5.16. Числа Фибоначчи (f_n) определяются формулами

$$f_0 = f_1 = 1; f_n = f_{n-1} + f_{n-2} \text{ при } n = 2, 3, \dots$$

а)* определить f — 40-е число Фибоначчи;

б) найти f — первое число Фибоначчи, большее m ($m > 1$);

в) вычислить s — сумму всех чисел Фибоначчи, которые не превосходят 1000.

5.17. Не используя стандартные функции (за исключением `abs`), вычислить с точностью $eps > 0$:

а)* $y = e^x = 1 + x/1! + x^2/2! + \dots + x^n/n! + \dots$;

б) $y = \sin x = x - x^3/3! + x^5/5! - \dots + x^{2n+1}/(2n+1)! + \dots$;

в) $y = \cos x = 1 - x^2/2! + x^4/4! - \dots + (-1)^n x^{2n}/(2n)! + \dots$;

г) $y = \ln(1+x) = x - x^2/2 + x^3/3 - \dots + (-1)^{n-1} x^n/n + \dots$ ($|x| < 1$);

д) $y = \arctg x = x - x^3/3 + x^5/5 - \dots + (-1)^n x^{2n+1}/(2n+1) + \dots$ ($|x| < 1$).

Считать, что требуемая точность достигнута, если очередное слагаемое по модулю меньше eps , — все последующие слагаемые можно уже не учитывать.

5.18. var k,i:integer; x,y:real;

Найти ошибки в следующем фрагменте программы:

а) $y := 0$; for $x := 0.1$ to 0.9 do $y := y + \sin(x)$;

б) $k := 81$; $y := 1$; for $i := 1$ to $\text{sqrt}(k)$ do $y := 2 * y$;

в) $k := 0$; for $i := 1$ to 9 do $k := k + \text{sqr}(i)$; $k := k * i$;

г) $k := 1$; for $i := 1$ to 64 do begin $i := 2 * i$; $k := k + i$ end

5.19*. Вычислить: $y = \sin 1 + \sin 1.1 + \sin 1.2 + \dots + \sin 2$.

5.20. Программа. Напечатать таблицу значений функций $\sin x$ и $\cos x$ на отрезке $[0,1]$ с шагом 0.1 в следующем виде (считать, что при печати на каждое вещественное значение отводится 5 знаков после запятой):

венное число отводится по 6 позиций строки):

x	$\sin(x)$	$\cos(x)$
0.0000	0.0000	1.0000
0.1000	0.0998	0.9950
...		
1.0000	0.8415	0.5403

5.21. Программа. Приближенно вычислить интеграл

$$\int_0^{\pi} \ln(2 + \sin x) dx,$$

используя формулу прямоугольников при $n=100$:

$$\int_a^b f(x) dx \approx h \cdot [f(x_1) + f(x_2) + \dots + f(x_n)],$$

где $h = (b-a)/n$, $x_i = a + ih - h/2$.

5.22*. Сколько раз будет выполняться тело следующего оператора цикла?

```
k:=0;  
for i:=1 to k+3 do k:=k+1
```

5.23*. Вычислить s —сумму квадратов всех целых чисел, попадающих в интервал $(\ln x, e^x)$, $x > 1$.

5.24. Вычислить k —количество точек с целочисленными координатами, попадающих в круг радиуса R ($R > 0$) с центром в начале координат.

5.25*. Если среди чисел $\sin x^n$ ($n = 1, 2, \dots, 30$) есть хотя бы одно отрицательное число, то логической переменной t присвоить значение *true*, а иначе—значение *false*.

а) Использовать оператор цикла с параметром и оператор перехода.

б) Не использовать оператор перехода.

5.26*. Логической переменной p присвоить значение *true*, если целое n ($n > 1$)—простое число, и значение *false* иначе.

5.27. Программа. Дано 100 вещественных чисел. Определить, образуют ли они возрастающую последовательность.

5.28. Программа. Дана последовательность из 70 целых чисел. Определить, со скольких отрицательных чисел она начинается.

5.29. Вычислить:

$$a) * p = \prod_{i,j=1}^{20} \frac{1}{i+j^2}; \quad b) s = \sum_{k=1}^{10} \frac{\sin kn}{k!}.$$

5.30*. Определить k — количество трехзначных натуральных чисел, сумма цифр которых равна n ($1 \leq n \leq 27$). Операции деления ($/$, div и mod) не использовать.

5.31. Напечатать в возрастающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр (операции деления не использовать).

5.32. Логической переменной t присвоить значение *true* или *false* в зависимости от того, можно ли представить натуральное число n в виде суммы трех полных квадратов.

5.33. Программа. Напечатать таблицу истинности для логической функции $F = (A \text{ and } B) \text{ or } \text{not}(B \text{ or } C)$ в следующем виде:

A	B	C	F
true	true	true	true
true	true	false	true
true	false	true	false
false	false	false	true

5.34. Программа. Даны вещественные числа c , d и ε ($c < d$, $\varepsilon > 0$). С точностью ε вычислить интеграл

$$\int_c^d \cos e^x dx,$$

используя формулу трапеций:

$$\int_a^b f(x) dx \approx I_n = h \cdot [f(a)/2 + f(a+h) + \\ + f(a+2h) + \dots + f(b-h) + f(b)/2],$$

где $h = (b-a)/n$.

Для обеспечения нужной точности воспользоваться следующим правилом Рунге: если приближенное значение интеграла I_n вычислять при $n = n_0, 2n_0, 4n_0, 8n_0$ и т. д.,

где n_0 — некоторое начальное число отрезков деления (например, $n_0=10$), тогда при $|I_{2n}-I_n|/3 < \varepsilon$ за искомую величину интеграла можно взять I_{2n} .

5.35. Программа. Дано $\varepsilon > 0$. С точностью ε найти корень уравнения

$$\pi x^3 - ex^2 + (2e+1)x + \pi^2 = 0.$$

5.36. Программа. Даны натуральное n и вещественные числа $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Рассматривая пары x_i, y_i как координаты точек на плоскости, определить радиус наименьшего круга (с центром в начале координат), внутрь которого попадают все эти точки.

5.37. Программа. Дано 80 вещественных чисел. Найти порядковый номер того из них, которое наиболее близко к какому-нибудь целому числу.

5.38. Программа. Дано 100 целых чисел. Определить, сколько из них принимает наибольшее значение.

5.39. Программа. Даны целое $n > 1$ и вещественные числа x_1, x_2, \dots, x_n . Вычислить:

$$M = \frac{\sum x_i}{n}, \quad D = \sqrt{\frac{\sum (x_i - M)^2}{n-1}}.$$

5.40. Программа. Даны непустая последовательность положительных вещественных чисел x_1, x_2, \dots, x_n (n заранее не известно), за которыми следует отрицательное число. Вычислить величину

$$\pi x_1 + (n-1)x_2 + \dots + 2x_{n-1} + x_n.$$

5.41. Программа. Даны целые числа x_1, x_2, \dots, x_{55} . Вычислить величину

$$x_1(x_2+x_3)(x_4+x_5+x_6)(x_7+x_8+x_9+x_{10}) \dots \\ \dots (x_{46}+x_{47}+\dots+x_{55}).$$

5.42. Программа. Даны последовательность из 100 целых чисел. Определить количество чисел в наиболее длинной подпоследовательности из подряд идущих нулей.

5.43. Программа. Дано не менее трех различных натуральных чисел, за которыми следует 0. Определить три наибольших числа среди них.

5.44. Программа. Дано 200 вещественных чисел. Определить, сколько из них больше своих «соседей», т. е. предыдущего и последующего чисел.

5.45. Программа. Даны непустая последовательность ненулевых целых чисел, за которой следует 0. Определить,

сколько раз в этой последовательности меняется знак.
(Например, в последовательности 1, -34, 8, 14, -5 знак меняется 3 раза.)

5.46. Программа. Даны целое $n > 2$ и вещественные числа $a_1, b_1, \dots, a_n, b_n$ ($a_i < b_i$). Рассматривая пары a_i и b_i как левые и правые концы отрезков на одной и той же прямой, определить концы отрезка, являющегося пересечением всех этих отрезков. Если такого отрезка нет, то сообщить об этом.

5.47. Программа. Найти все целые корни уравнения $ax^3 + bx^2 + cx + d = 0$, где a, b, c и d —заданные целые числа, причем $a \neq 0$ и $d \neq 0$. (Замечание: целыми корнями могут быть только положительные и отрицательные делители коэффициента d .)

5.48. Программа. Определить, является ли заданное натуральное число совершенным, т. е. равным сумме всех своих (положительных) делителей, кроме самого этого числа (например, число 6 совершенно: $6 = 1 + 2 + 3$).

5.49. Программа. Дано 10 натуральных чисел. Найти их наибольший общий делитель.

5.50. Программа. Дано целое $n > 2$. Напечатать все простые числа из диапазона $[2, n]$.

5.51. Программа. Напечатать все простые делители заданного натурального числа.

5.52. Программа. Данна последовательность из не менее чем двух натуральных чисел, за которой следует 0. Вычислить сумму тех из них, порядковые номера которых—простые числа.

5.53. Программа. Даиа непустая последовательность натуральных чисел, за которой следует 0. Вычислить сумму тех из них, порядковые номера которых—числа Фибоначчи (см. 5.16).

5.54. Программа. Найти сумму цифр заданного натурального числа.

5.55. Программа. Определить число, получаемое выписыванием в обратном порядке цифр заданного натурального числа.

5.56. Программа. Определить, является ли заданное натуральное число палиндромом, т. е. таким, десятичная запись которого читается одинаково слева направо и справа налево.

5.57. Программа. Дано натуральное k . Напечатать k -ю цифру последовательности

а) 12345678910111213..., в которой выписаны подряд все натуральные числа;

- б) 149162536..., в которой выписаны подряд квадраты всех натуральных чисел;
в) 1123581321..., в которой выписаны подряд все числа Фибоначчи.

6. ЛИТЕРНЫЙ ТИП

6.1*. Всегда ли справедливо следующее утверждение:

- а) если c —литера и ' $0' \leq c \leq '9'$ ', то c —цифра;
б) если c —литера и ' $'a' \leq c \leq 'z'$ ', то c —строчная латинская буква;

в) ' $9' < 'a'$ ';

г) ' $'0' = 0$ ';

д) $\text{ord}('0') = 0$;

е) ' $a' = 'A'$ ';

ж) если c и d —литеры, то $c < d$ тогда и только тогда, когда $\text{ord}(c) < \text{ord}(d)$;

з) $\text{ord}(\text{chr}(k)) = k$ для любого целого k ;

и) $\text{chr}(\text{ord}(c)) = c$ для любой буквы c .

6.2*. Вычислить значения выражений:

а) $\text{pred}('7')$; б) $\text{succ}('0') = \text{pred}('2')$;

в) $\text{ord}('5') - \text{ord}('0')$; г) ' $d' > 'a'$ ';

д) $\text{ord}('q') < \text{ord}('z')$; е) $\text{chr}(25) >= \text{chr}(16)$;

ж) $\text{ord}(\text{pred}(\text{chr}(24)))$; з) $\text{chr}(\text{ord}('5') + 1)$.

6.3*. Имеется литерная переменная d . Присвоить логической переменной t значение *true*, если выполняется указанное условие, и значение *false* в противном случае:

а) значение d —это литера '*' ;

б) значение d —литера 'a' или 'q' ;

в) значение d —цифра.

6.4*. Определить значение литерной переменной d после выполнения следующих операторов:

а) $c := '+'$; б) $c := '+'$; $d := 'c'$

6.5*. Вычислить s — сумму порядковых номеров всех букв, входящих в слово SUM.

6.6*. Напечатать текст, образованный литерами с порядковыми номерами 65, 71 и 69.

6.7*. Литерой переменной *next* присвоить цифру, следующую за цифрой, являющейся значением литерой переменной *dig*, считая при этом, что за '9' следует '0'.

6.8*. Логической переменной b присвоить значение *true*, если между литерами 'a' и 'z' нет иных символов, кроме строчных латинских букв, и значение *false* иначе.

6.9*. Напечатать в одну строку все буквы между 'A' и 'Z', включая эти буквы.

6.10. Напечатать таблицу следующего вида:

a)*	100...00	б) 999...99	в) 0123456789
	020...00	088...88	1234567890

	000...09	000...01	9012345678

Далее в этом параграфе под «текстом» понимается заданная во входном файле *input* последовательность литер (возможно, пустая), за которой следует точка (в сам текст точка не входит).

6.11*. Программа. Напечатать *true*, если в заданном тексте буква а встречается чаще, чем буква б, и напечатать *false* в противном случае.

6.12. Программа. Если в заданный текст входит каждая из букв слова key, тогда напечатать yes, иначе—no.

6.13. Программа. Проверить, правильно ли в заданном тексте расставлены круглые скобки (т. е. находится ли справа от каждой открывающей скобки соответствующая ей закрывающая скобка, а слева от каждой закрывающей—соответствующая ей открывающая). Ответ—ДА или НЕТ.

6.14*. Программа. Определить, является ли заданный текст правильной записью целого числа (возможно, со знаком).

6.15. Известно, что в заданный текст входит буква а, причем не на последнем месте. Требуется напечатать литеру текста, непосредственно следующую за первым вхождением а. Можно ли решить эту задачу следующим образом (*c*—литерная переменная):

repeat read(c) until c=='a'; writeln(succ(c))?

6.16. Программа. Напечатать заданный непустой текст:
а)* удалив из него все цифры и удвоив знаки «+» и «—»;

б)* удалив из него все знаки «+», непосредственно за которыми идет цифра;

в) удалив из него все буквы б, непосредственно перед которыми находится буква с;

г) заменив в нем все пары ph на букву f.

6.17. Программа. Напечатать заданный текст, удалив из него лишние пробелы, т. е. из нескольких подряд идущих пробелов оставить только один.

6.18. Программа. Заданный текст распечатать по строкам, понимая под строкой либо очередные 60 литер, если среди них нет запятой, либо часть текста до запятой включительно.

6.19. Программа. Даны непустые последовательности из латинских букв; соседние слова отделены друг от друга запятой, за последним словом — точка. Определить количество слов, которые:

- а)* начинаются с буквы а;
- б) оканчиваются буквой w;
- в) начинаются и оканчиваются одной и той же буквой;
- г) содержат хотя бы одну букву d;
- д) содержат ровно три буквы e.

6.20*. Значениями литерных переменных $c2$, $c1$ и $c0$ являются цифры. Присвоить целой переменной k число, составленное из этих цифр (например, если $c2='8'$, $c1='0'$ и $c0='5'$, то $k=805$).

6.21*. Присвоить литерным переменным $c2$, $c1$ и $c0$ соответственно левую, среднюю и правую цифры трехзначного числа k .

6.22*. Используя только литерный ввод, т. е. процедуру $read(c)$, где c — литерная переменная, ввести непустую последовательность цифр, перед которой может находиться знак «+» или «—» и за которой следует пробел, и, получив соответствующее целое число, присвоить его целой переменной k .

6.23. Используя только литерный вывод, т. е. процедуру $write(c)$, где c — литерный параметр, вывести на печать значение целой переменной k (знак «+» не печатать).

6.24. Программа. Дано натуральное число n . Напечатать в троичной системе счисления целые числа от 0 до n .

6.25. Программа. Даны непустые последовательности неотрицательных целых чисел, записанных в семеричной системе счисления; между соседними числами — пробел, за последним — точка. Напечатать в десятичной системе наибольшее из них.

6.26. Программа. Задано неотрицательное целое число, записанное в восьмеричной системе счисления (за числом — пробел). Напечатать это число в пятеричной системе счисления.

6.27. Программа. Даны последовательность литер, имеющая следующий вид: $d_1 \pm d_2 \pm \dots \pm d_n$ (d_i — цифры, $n > 1$), за которой следует точка. Вычислить значение этой алгебраической суммы.

6.28. Используя только литерный ввод, ввести заданное вещественное число (за ним — пробел), записанное по правилам языка Паскаль, и присвоить его вещественной переменной x .

6.29. Используя только литерный вывод, напечатать вещественное число x в следующей форме:

$$\pm 0.d_1d_2\dots d_n E \pm p_1p_2,$$

где d_i, p_j — цифры, причем $d_1 \neq 0$, если $x \neq 0$.

6.30*. Имеются целые числа k и n от 1 до 80. Напечатать в очередной строке (номера ее позиций — от 1 до 80):

а) литеру * в k -й позиции;

б) литеру * в k -й позиции и литеру I в n -й позиции (при $k=n$ печатать только звездочку).

6.31. Программа. Напечатать график функции $y=x^3-1$ на отрезке $[-1, 2]$ с шагом 0.1. Ось ОХ направить по вертикали вниз, а ось ОY — по горизонтали вправо. В каждой строке печатать «кусочек» оси ОХ (например, литеру I) и звездочку — в позиции, соответствующей очередному значению функции; ось ОY не печатать.

Примерный вид графика:

*
* I
* I
* I
*
I *
I .

6.32. Программа. Напечатать в точках 1, 2, ..., k , где k — заданное целое число от 2 до 70, график функции Эйлера $\phi(n)$, вычисляющей количество целых чисел от 1 до $n-1$, взаимно простых с числом n .

6.33. Программа. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству:

а) текст является десятичной записью числа, кратного 9;

б) текст является записью четного числа в семеричной системе;

в) текст является десятичной записью числа, кратного 6;

г) текст является десятичной записью числа, кратного 4;

д) текст является шестнадцатеричной записью числа, кратного 5;

е) текст начинается с некоторой ненулевой цифры, за которой следуют только буквы, и их количество равно числовому значению этой цифры;

ж) текст начинается с k букв ($1 \leq k \leq 9$), за которыми

следует только одна литература—цифра с числовым значением k ;

з) текст совпадает с начальным отрезком ряда 0123456789 (например: 0, 01, 012);

и) текст совпадает с конечным отрезком ряда 0123456789 (например: 9, 89, 789);

к) текст совпадает с каким-то отрезком ряда 0123456789 (например: 2, 678);

л) текст состоит только из цифр, причем их числовые значения образуют арифметическую прогрессию (например: 2468, 741, 3);

м) текст содержит (помимо букв) только одну цифру, причем ее числовое значение равно длине текста;

н) сумма числовых значений цифр, входящих в текст, равна длине текста.

7. ПЕРЕЧИСЛИМЫЕ И ОГРАНИЧЕННЫЕ ТИПЫ.

ОПЕРАТОР ВАРИАНТА

7.1*. Имеются описания

type сезон=(зима, весна, лето, осень);

var x,y:сезон;

т:(тепло, холодно);

Ответить на следующие вопросы.

а) Какие значения могут принимать переменные x , y и t ? Допустимы ли присваивания:

1) $x:=$ весна; 2) $y:=x$; 3) $t:=$ тепло;

4) $y:=t$; 5) $t:=$ жарко?

б) Вычислить значения выражений:

1) весна<лето; 2) зима<=лето; 3) осень<зима;

4) весна <> тепло; 5) succ(весна); 6) pred(весна);

7) succ(осень); 8) pred(холодно)

в) Вычислить значения выражений:

1) ord(весна); 2) ord(осень)+ord(холодно)

г) Допустим ли оператор цикла с заголовком

for x:=весна to осень do?

д) Допустимы ли следующие операции ввода-выводан

1) read(x); 2) write(лето);

3) writeln('зимой ',t)?

7.2*. Указать ошибки в следующем разделе типов:

type буква=('а','б','с','д');

гласная=(а,е,и,о,у);

согласная=(б..д, ф, г);
карта=(6, 7, 8, 9, 10, валет, дама, король, туз);
корень=(1.00, 1.41, 1.73, 2.00);
фигура=(слон, конь, ладья, ферзь);
конь=(пегий, сивый, каурый);
животное=(лев, слои, жираф);
деление=(div, mod);
лог=boolean;

7.3*. Имеются описания

```
type digit='0'..'9';
var d:digit; k:0..9; n:integer;
```

Ответить на следующие вопросы.

а) Каковы значения типа digit? Каков базовый тип для этого ограниченного типа? Допустимы ли присваивания:

- 1) $d := '7'$; 2) $d := 'a'$; 3) $d := 7$?

б) Какие значения может принимать переменная k? Каков ее базовый тип? Допустимы ли присваивания:

- 1) $k := 5$; 2) $k := 10$; 3) $k := -0$; 4) $k := '5'$;
- 5) $k := \text{ord}(d)$; 6) $k := \text{ord}(d) - \text{ord}('0')$?

в) Верно ли, что к значениям ограниченного типа можно применять те же операции, что и к значениям базового типа? Есть ли ошибки в операторе

```
if k+n>7*k then k:=abs(n) mod 10
else d:=chr(k+ord('0'))?
```

г) Можно ли в Паскале определить ограниченный тип, состоящий из простых чисел?

7.4*. Указать ошибки в следующих описаниях:

```
const pi=3.14159;
type цифра='0'..'9';
      буква=a..z;
      угол=-pi..pi;
      отрезок=0..n-1;
      период=-pi..pi;
      плюс='+'..'+';
      неделя=(вс,пн,вт,ср,чт,пт,сб);
      будни=пн..пт;
      выходной=сб..вс;
```

7.5*. type месяц=(янв, фев, мар, апр, май, июн, июл, авг,
сен, окт, ноя, дек);

день=1..31;

```
var d1,d2:день; m1,m2:месяц; t:boolean;
```

Перемениой *t* присвоить значение *true*, если дата *d1*, *m1* предшествует (в рамках года) дате *d2*, *m2*, и значение *false* иначе.

7.6*. var *m,m1:месяц*; {определение «месяц» см. в 7.5}
k:1..maxint; n:1..12;

Присвоить переменной *m1*:

а) название месяца, следующего за месяцем *m* (с учетом того, что за декабрем идет январь);

б) название *k*-го месяца после месяца *m*;

в) название *n*-го месяца года.

7.7. type страна=(Австрия,Болгария,Греция,Италия,
Норвегия,Франция,ФРГ);
столица=(Вена,София,Афины,Рим,Осло,
Париж,Бони);

var *st:страна; cap:столица;*

По значению переменной *st* (названию страны) присвоить переменной *cap* название столицы этой страны.

7.8. var *P: (ада, бейсик, модула2, лисп,
паскаль, пл1, фортран);
A: (ada, basic, modula2, lisp,
pascal, pl1, fortran);*

По *P*—русскому названию языка программирования присвоить переменной *A* английское название этого языка.

7.9. type название=(иоль, один, два, три,
четыре, пять);
var: *d:'0'..'5'; n:название;*

По литере-цифре *d* присвоить переменной *n* название этой цифры.

7.10. type нота=(до,ре,ми,фа,соль,ля,си);
интервал=(секунда, терция, квarta,
квинта, секста, септима);
var *n1,n2: нота; i:интервал;*

Определить *i*—интервал, образованный нотами *n1* и *n2* (*n1* \neq *n2*): секунда—это интервал из двух соседних (по кругу) нот (например, ре и ми, си и до), терция—интервал через ноту (например, фа и ля, си и ре) и т. д.

7.11*. Определить, какими будут значения переменных *p* и *d* после выполнения операторов

*p:=true; d:=1;
case k mod 10 of*

```
3, 2, 7, 5; d:=k;  
1;  
4,8; begin p:=false; d:=2 end;  
9,6; begin p:=false; d:=3 end  
end {of case}
```

если целая переменная *k* имеет следующее значение:

- a) 6; б) 235; в) 71; г) 100.

7.12. var u,w:'a'..'z';

Найти ошибки в следующем операторе:

```
case u of  
  'a'..'w'; w:=succ(u);  
  'u', 'y'; u:='g'; w:=u  
end
```

7.13*. type сезон=(зима,весна,лето,осень);
месяц=(янв,фев,мар,апр,май,июн,
июл,авг,сен,окт,ноя,дек);
var т:месяц; s:сезон;

Определить *s*—сезон, на который приходится месяц *т*.

7.14. type страна=(ГДР,Куба,Лаос,Монако,Непал,
Польша);
континент=(Азия,Америка,Европа);
var s:страна; c:континент;

По *s*—названию страны определить *c*—название ее континента.

7.15. type единица=(дециметр,километр,метр,
миллиметр,сантиметр);
длина=real;
var x:длина; р:единица;

Значение переменной *x*, означающее некоторую длину
в единицах *p*, заменить на величину этой же длины в метрах.

7.16*. type цвет=(черный,серый,белый);
var с:цвет;

Напечатать значение переменной *c*.

7.17. var k:1..9;

Напечатать значение переменной *k* римскими цифрами.

7.18*. type letter=(a,b,c,d);
var x:letter;

Ввести заданное во входном файле значение типа *letter*
(т. е. a, b, c или d) и присвоить его переменной *x*.

7.19. Для целого числа k от 1 до 99 напечатать фразу «мне k лет», учитывая при этом, что при некоторых значениях k слово «лет» надо заменить на слово «год» или «года».

7.20. Для натурального числа k напечатать фразу «мы нашли k грибов в лесу», согласовав окончание слова «гриб» с числом k .

7.21. type падеж=(им,род,дат,внн,твор,предл);
слово=(степь,боль,тетрадь,дверь);
var w:слово; p:падеж;

Напечатать слово w в падеже p в единственном числе (например, при $w=степь$ и $p=твор$ надо напечатать слово *степью*).

7.22. type курс=(С,В,Ю,З); {север,восток,юг,запад}
приказ=(вперед,вправо,назад,влево);
var K1,K2:курс; PR:приказ;

Корабль сначала шел по курсу $K1$, а затем его курс был изменен согласно приказу PR . Определить $K2$ —новый курс корабля.

7.23*. var d:28..31; t:месяц; {см. 7.5}

Переменной d присвоить количество дней в месяце t (год считать невисокосным).

7.24. var y:1901..2000; t:месяц; {см. 7.5} d:1..31;
t:boolean;

Переменной t присвоить значение *true*, если тройка y , t , d образует правильную дату, и значение *false*—иначе (при 31 июня и т. п.).

7.25. var d,d1:1..31; m,m1:месяц; {см. 7.5}
y:1901..2000; y1:1901..2001;

По дате d , m , y определить $d1$, $m1$, $y1$ —дату следующего дня.

7.26. var k:1..366; d:1..31; t:месяц; {см. 7.5}

а)* определить k —порядковый номер того дня високосного года, который имеет дату d , t ;

б) определить d , t —дату k -го по счету дня високосного года.

7.27. type число=1..31;
месяц=(янв,фев,мар,апр,май,июн,июл,
авг,сен,окт,ноя,дек);
деньнедели=(вс,пн,вт,ср,чт,пт,сб);
var d:число; t:месяц; wd1,wd: деньнедели;
k:0..12;

Считая, что год невисокосный и его 1 января приходится на день недели $wd1$, определить:

а) wd —день недели, на который приходится день с датой d, m ;

б) k —количество понедельников в году, приходящихся на 13-е числа.

7.28. Найти все ошибки в следующих программах и объяснить, какие правила языка Паскаль при этом нарушены:

а) program ошибки (input,output);

```
type месяц=(янв,фев,мар,апр,май,июн,июл,
авг,сен,окт,ноя,дек);
```

```
зима=дек..фев;
```

```
весна=мар..май;
```

```
var m:месяц; k:1..12;
```

```
begin read(m);
```

```
if m>весна then m:=июнь;
```

```
for k:=ord(янв) to ord(m) do
```

```
    m:=succ(m);
```

```
    writeln(m)
```

```
end.
```

б) program ошибки(input,output)

```
type цифра='0','9';
```

```
знак='+','-','*','/');
```

```
var d:цифра; t:boolean;
```

```
begin read(d);
```

```
    case d of
```

```
        2,3,5,7: t:=true; d:=succ(d);
```

```
        0,1,4,6,8,9: t:=false;
```

```
        writeln(t,d)
```

```
end.
```

7.29. В старояпонском календаре был принят 60-летний цикл, состоявший из пяти 12-летних подциклов. Подциклы обозначались названиями цвета: зеленый, красный, желтый, белый и черный. Внутри каждого подцикла годы носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. (1984 год—год зеленой крысы—был началом очередного цикла).

Написать программу, которая вводит номер некоторого года нашей эры и печатает его название по старояпонскому календарю.

7.30. Программа. Пусть значение функции $f(n)$ равно количеству букв в записи числа n русскими словами

$f(1)=4$ (один), $f(3)=3$ (три), $f(42)=8$ (сорок два) и т. п.
Напечатать все натуральные числа n , меньшие 100, для которых $f(n)=n$.

8. РЕГУЛЯРНЫЕ ТИПЫ: ВЕКТОРЫ

8.1*. Имеются описания:

```
type день=(вчера,сегодня,завтра);
    вектор=аттаг [1..30] of real;
var a:вектор;
    b:packed аттаг [-2..2] of (x,y,z);
    c:array ['0'..'9'] of вектор;
    d:array [день] of 0..23;
```

Для каждого из массивов a , b , c и d указать:

- сколько в нем элементов;
- какие значения могут принимать его элементы;
- как указать его первый и последний элементы.

8.2*. Описать регулярный тип R , объединяющий в себе массивы, элементами которых являются натуральные числа, а индексами—любые литеры.

8.3. Ответить на следующие вопросы.

- Может ли массив содержать один элемент? А ии одного?
- Можно ли во время выполнения программы изменить размер массива (количество элементов в нем)?
- Могут ли элементами некоторого массива быть числа 1, 1.41, 1.73 и 2?
- Верно ли, что тип элементов массива может быть любым?
- Может ли типом индекса массива быть тип *integer* или *real*?
- Каковы достоинства и недостатки упакованных массивов по сравнению с неупакованными? Кто (автор программы или транслятор) определяет, упаковывать массив или нет?

8.4*. Указать ошибки в следующих описаниях:

```
const n=50;
type слово=packed array [0..n-1] of буква;
    буква='a'..'z';
    вектор=аттаг [real] of integer;
    цифры=аттаг [true..false] of (1, 2, 3, 4);
var k:1..maxint;
    x:аттаг [1..k] of char;
```

у:packed array [—п..п] of 0..0;
з:array [(a,b,c)] of boolean;
8.5*. const n=41;
var x:array [1..n] of real; y:real;

Написать фрагмент программы для вычисления:

- $y = \sqrt[n]{|x_1 \cdot x_2 \cdot \dots \cdot x_n|};$
- $y = \max_i x_i;$
- $y = x_1 - x_2 + x_3 - \dots - x_{n-1} + x_n;$
- $y = x_1 x_n + x_2 x_{n-1} + \dots + x_n x_1;$
- $y = x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2;$
- $y = x_n (x_n + x_{n-1}) (x_n + x_{n-1} + x_{n-2}) \dots (x_n + \dots + x_1).$

8.6. const m=50;
var C: packed array [0..m] of char;
B: packed array [0..m] of boolean;

По массиву *C* получить массив *B*: элементу *B[i]* присвоить *true*, если *C[i]* — цифра, и присвоить *false* иначе.

8.7. type имя=(Валя, Гена, Женя, Коля, Маша, Нина,
Саша, Таня, Федя, Шура);
var Пол:array [имя] of (муж,жен);
Рост:array [имя] of 140..200;
И:имя; Ср:real;

По массивам *Пол* и *Рост* определить:

- И* — имя самого рослого мужчины;
 - Ср* — средний рост женщины.
- 8.8*. type текст=packed array [1..72] of char;
шифр=packed array [char] of char;
var t:текст; k:шифр;

Зашифровать текст *t*, заменив в нем каждую литеру на значение элемента массива *k*, индексом которого является эта литера.

8.9. type деньнедели=(пи,вт,ср,чт,пт,сб,вс);
var год: array [1..365] of деньнедели;

Присвоить каждому элементу *год[i]* название того дня недели, на который приходится *i*-й по счету день невисокосного года, если известно, что 1 января — среда (*год[1]:=ср*, *год[2]:=чт* и т. д.).

8.10. type республика=(Грузия,Россия,Украина);
город=(Киев,Москва,Одесса,Сочи,
Тбилиси,Томск);
var x:array [1..20] of город;

Напечатать название республики, города которой наиболее часто встречаются в массиве x (считать, что такая республика одна).

8.11. Указать, какие операции над массивами (как едиными объектами) допустимы в Паскале, и найти ошибки в следующей программе:

```
program ошибки (input,output);
var x,y:array [1..20] of real;
z,u:array [1..50] of real;
i:integer;
begin read(x,y);
if x<>y then begin z:=x;
x:=y; y:=z end
else x:=x+y;
for i:=1 to 20 do u[i]:=x[i]+y[i];
z:=u;
writeln(x)
end.
```

8.12*. const n=30;
type вектор=array [1..n] of integer;
var a,b,c:вектор;

Если векторы a и b различны, то вектору c присвоить их сумму, иначе в вектор c переписать элементы массива a .

8.13*. Программа. Дано 100 целых чисел. Распечатать их в обратном порядке по 6 чисел в строке.

8.14*. var x:array [1..40] of char;
y:array [0..39] of char;

Переписать элементы массива x в массив y . (Можно ли эту задачу решить с помощью оператора $y:=x$?)

8.15*. Для решения каких из следующих задач нужны массивы, а в каких задачах можно обойтись и без них?

а) Дано 50 чисел. Найти их среднее арифметическое.

б) Дано 50 чисел. Определить, сколько среди них отличных от последнего числа.

в) Дано 100 чисел. Напечатать сначала все отрицательные из них, а затем все остальные.

г) Дано число a . Определить первый отрицательный член последовательности x_1, x_2, x_3, \dots , где $x_1=a, x_n=\lg(x_{n-1})$.

8.16*. Программа. Вычислить величину

$$(x_1y_1+x_2y_2+\dots+x_{20}y_{20})/(x_2y_2+x_4y_4+\dots+x_{20}y_{20}),$$

если числа для ввода заданы в следующем порядке:

а) $x_1, x_2, \dots, x_{20}, y_1, y_2, \dots, y_{20}$;

б) $x_1, y_1, x_2, y_2, \dots, x_{20}, y_{20}$.

8.17. Программа. По заданным вещественным числам $a_0, a_1, \dots, a_{20}, t$ вычислить значение многочлена

$$a_{20}x^{20} + a_{19}x^{19} + \dots + a_1x + a_0$$

и его производной в точке t .

Какие изменения надо внести в программу, чтобы она правильно выполнялась для многочлена 45-й степени?

8.18*. Программа. Дан текст из 80 литер. Напечатать сначала все цифры, входящие в него, а затем все остальные буквы, сохраняя при этом взаимное расположение литер в каждой из этих двух групп.

8.19. Программа. Дан текст, содержащий от 1 до 70 букв, за которым следует точка. Напечатать этот текст в обратном порядке.

8.20. Программа. Дан непустой текст из цифр, за которым следует точка. Напечатать цифру, наиболее часто встречающуюся в этом тексте (если таких цифр несколько, напечатать любую из них).

8.21. const n=1000;

var s:packed array [1..n] of char;

Напечатать те элементы массива s , индексы которых являются:

а)* степенями двойки (1, 2, 4, 8, 16, ...);

б) полными квадратами (1, 4, 9, 16, 25, ...);

в) числами Фибоначчи (1, 2, 3, 5, 8, 13, ...).

8.22. var x:array [1..100] of real;

 a:array [1..30] of 1..100;

 s:real;

Вычислить s — сумму тех элементов массива x , индексы которых совпадают со значениями элементов массива a ($a_i \neq a_j$ при $i \neq j$).

8.23. var x:array [1..9999] of real; s:real;

Вычислить (индекс 1-го слагаемого каждой суммы — квадрат):

$$s = (x_1 + x_2 + x_3)(x_4 + x_5 + \dots + x_8)(x_9 + \dots + x_{15}) \dots$$

$$\dots (x_{9961} + \dots + x_{9999}).$$

8.24. const n=20;

var s:packed array [1..n] of char;

Напечатать литеры s_i массива s в виде таблицы:

$s_1 s_2 s_3 \dots s_{n-1} s_n$

$s_2 s_3 s_4 \dots s_n s_1$

...

$s_n s_1 s_2 \dots s_{n-1} s_{n-2}$

8.25. Программа. Напечатать величины a_0, a_1, \dots, a_{99} , где a_0 — заданное целое число, $a_n = a_{(n/2)} + a_{n-1}$ при $n=1, 2, \dots, 99$.

8.26. const n=100;

var x:array [1..n] of real;

Преобразовать массив x по следующему правилу (x'_k — значение k -го элемента массива после преобразования):

а) $x'_k = \max x_i$ при $1 \leq i \leq k$;

б)* элементы массива расположить в обратном порядке;

в) $x'_1 = x_1, x'_n = x_n, x'_k = (x_{k-1} + x_k + x_{k+1})/3$ при $k=2, 3, \dots, n-1$;

г)* элементы массива циклически сдвинуть на одну позицию влево: $x'_n = x_1, x'_k = x_{k+1}$ при $k=1, 2, \dots, n-1$;

д) элементы массива циклически сдвинуть на две позиции влево.

8.27. var x,y:array [1..70] of real;

k:1..69;

Преобразовать массив x по следующему правилу (воспользовавшись массивом y как вспомогательным):

а) все отрицательные элементы массива x перенести в его начало, а все остальные — в конец, сохраняя исходное взаимное расположение как среди отрицательных, так и среди остальных элементов;

б) элементы массива x циклически сдвинуть на k позиций влево.

8.28*. const k=50; m=20; n=70; {n=k+m}

var x:array [1..k] of real;

y:array [1..m] of real;

z:array [1..n] of real;

Элементы каждого из массивов x и y упорядочены по неубыванию. Объединить элементы этих двух массивов в один массив z так, чтобы они снова оказались упорядоченными по неубыванию.

8.29. var k:0..99999;

d:packed array [1..5] of '0'..'9';

а) В массив d записать цифры числа k .

б) Получить k — целое, составленное из цифр массива d .

8.30. type мантисса=packed array [1..9]

of '0'..'9';

порядок=packed array [1..2]

of '0'..'9';

var mантисса; порядок; x:real;

Переменной x присвоить вещественное число $0.m_1m_2\dots m_n \cdot 10^{p_1p_2}$.

8.31. type месяц=(янв, фев, мар, апр, май, июн, июл,
авг, сен, окт, ноя, дек);

var КД:array [месяц] of 28..31;

Присвоить каждому элементу массива KD значение, равное количеству дней в соответствующем месяце (невисокосного) года.

8.32. var t:array [1..365] of real;
 т:месяц; {см. 8.31}

По массиву t , где указана температура каждого дня некоторого невисокосного года, определить m — название месяца с наибольшей среднемесячной температурой.

8.33. const n=40;
 var x:array [1..n] of integer;
 y,k:integer; t:boolean;

Написать фрагмент программы для решения следующей задачи:

а)* переменной t присвоить значение *true*, если элементы массива x упорядочены строго по возрастанию, и значение *false* иначе;

б) переменной t присвоить значение *true*, если в массиве x нет нулевых элементов и при этом положительные элементы чередуются с отрицательными, и значение *false* иначе;

в) переменной k присвоить либо номер первого вхождения y в массив x , либо число $n+1$, если y не входит в x ;

г) вычислить $y=x_1+x_1x_2+x_1x_2x_3+\dots+x_1x_2\dots x_m$, где m — либо номер первого отрицательного элемента массива x , либо число n , если в массиве x нет отрицательных элементов.

8.34. Программа. Дан текст из 80 литер. Определить, симметричен ли он, т. е. читается ли он одинаково слева направо и справа налево.

8.35*. type фамилия=(Бетелин, Войскунский, Коннов,
Мальковский, Пильщиков,
Школьный);

имя=(Василий, Владимир, Игорь, Михаил,
Олег, Юрий);

var MM511:array [фамилия] of имя;
ЕстьТезки:boolean;

Переменной *ЕстьТезки* присвоить значение *true*, если в массиве *ММ511* есть одинаковые имена, и значение *false* иначе.

8.36. var x:array [1..50] of 1.. maxint;
t:boolean;

Переменной *t* присвоить значение *true* или *false* в зависимости от того, есть или нет среди элементов массива *x*:

- а) хотя бы одно число Фибоначчи;
- б) не менее двух степеней двойки.

8.37. type слово=packed array [1..10] of char;
var x,y:слово; eq:boolean;

Считая, что в каждом из слов *x* и *y* нет повторяющихся литер, присвоить переменной *eq* значение *true*, если слова *x* и *y* отличаются только порядком входящих в них литер, и значение *false* иначе.

8.38. const n=100;
var x:array [1..n] of real;

Упорядочить массив *x* по неубыванию (т. е. переставить его элементы так, чтобы для всех *k* выполнялось $x_k \leq x_{k+1}$), используя следующий алгоритм сортировки (упорядочения):

а)* *сортировка выбором*: отыскивается максимальный элемент и переносится в конец массива; затем этот метод применяется ко всем элементам, кроме последнего (он уже находится на своем окончательном месте), и т. д;

б) *сортировка обменом (метод пузырька)*: последовательно сравниваются пары соседних элементов x_k и x_{k+1} ($k=1, 2, 3, \dots, n-1$) и, если $x_k > x_{k+1}$, то они переставляются; тем самым наибольший элемент оказывается на своем месте в конце массива; затем этот метод применяется ко всем элементам, кроме последнего, и т. д;

в) *сортировка вставками*: пусть первые *k* элементов массива уже упорядочены по неубыванию; берется (*k*+1)-й элемент и размещается среди первых *k* элементов так, чтобы упорядоченнымми оказались уже *k*+1 первых элементов; этот метод применяется при *k* от 1 до *n*-1.

8.39*. const n=500;
var x:array [1..n] of integer;
p:integer; k:0..n;

Элементы массива *x* упорядочены по возрастанию. Требуется присвоить переменной *k* номер элемента массива *x*, равного числу *y*, или 0, если такого элемента нет.

Использовать следующий метод двоичного (бинарного) поиска: сравнить *p* со средним элементом массива (или

элементом около середины); если эти числа равны, поиск завершается, если же p меньше среднего элемента, то p надо искать в левой половине массива, а иначе—в правой; к выбранной половине примениается этот же алгоритм.

8.40. const $n=80$;

type цифра=0..9;

число=packed array [1..n] of цифра;

var a,b,c:число; t:boolean;

Рассматривая a и b как последовательности цифр десятичной записи некоторых неотрицательных целых чисел, получить c —аналогичное представление для суммы этих двух чисел. Если в сумме окажется более n цифр, то ее левую цифру отбросить, а переменной t присвоить значение *true*, иначе переменной t присвоить значение *false*.

8.41. type элемент=0..99;

множество=packed array [элемент]
of boolean;

var x,y,z:множество; t:boolean;

Рассматривая массивы x , y и z как представления некоторых множеств из объектов типа *элемент* ($x[k]=true$, если элемент k принадлежит множеству x , и $x[k]=false$ иначе, и т. п.), реализовать следующие операции над этими массивами-множествами:

а) переменной t присвоить значение *true*, если множество x является подмножеством множества y , и значение *false* иначе;

б) $z=x \cap y$ —пересечение множеств;

в) $z=x \cup y$ —объединение множеств;

г) $z=x \setminus y$ —разность множеств (в z входят все элементы из x , которые не входят в y).

8.42. const $n=20$; $n1=21$; $\{n1=n+1\}$

var P,Q:array [0..n] of real;

R:array [0..n1] of real;

a:real;

По P —массиву коэффициентов многочлена

$$P(x)=p_0x^n+p_1x^{n-1}+\dots+p_{n-1}x+p_n$$

получить:

а)* R —массив коэффициентов многочлена $(x-a)P(x)$;

б) Q —массив коэффициентов многочлена $P(x+a)$.

8.43. Программа. Даана последовательность из 100 различных целых чисел. Найти сумму чисел этой последовательности, расположенных между максимальным и минимальным числами (в сумму включить и оба этих числа).

8.44. Программа. Даны координаты n точек на плоскости: $x_1, y_1, \dots, x_n, y_n$ ($n=20$). Найти номера двух точек, расстояние между которыми наибольшее (считать, что такая пара точек единственная).

8.45. Программа. Даны две последовательности по 30 целых чисел в каждой. Найти наименьшее среди тех чисел первой последовательности, которые не входят во вторую последовательность (считая, что хотя бы одно такое число есть).

8.46. Программа. Данна последовательность из 20 целых чисел. Определить количество инверсий в этой последовательности (т. е. таких пар элементов, в которых большее число находится слева от меньшего: $x_i > x_j$, при $i < j$).

8.47. Программа. Дан текст из строчных латинских букв, за которым следует точка. Напечатать в алфавитном порядке все буквы, которые входят в этот текст по одному разу.

8.48. Программа. Напечатать заданный текст из 100 литер, удалив из него повторные вхождения каждой линтеры.

8.49. Программа. Определить, сколько различных литер входит в заданный текст, содержащий не более 100 литер и оканчивающийся точкой (в сам текст точка не входит).

8.50. Программа. Даана непустая последовательность слов из строчных латинских букв; между соседними словами — запятая, за последним словом — точка. Напечатать все буквы, которые входят в наибольшее количество слов этой последовательности.

8.51. Программа. Подсчитать количество «счастливых» шестизначных автобусных билетов, т. е. таких, в номерах которых сумма трех первых цифр равна сумме трех последних. (Воспользоваться тем, что число «счастливых» билетов равно $s_0^3 + s_1^3 + \dots + s_{99}^3$, где s_n — количество чисел от 0 до 999, сумма цифр которых равна n .)

8.52. Программа. Дано целое k от 2 до 20. Найти коэффициенты k -го многочлена Чебышева. (Замечание: многочлены Чебышева $T_n(x)$ определяются формулами

$$T_0(x) = 1; T_1(x) = x; \\ T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n = 2, 3, \dots$$

8.53. Программа. Даны вещественные числа a_0, a_1, \dots, a_{15} . Найти коэффициенты многочлена $(x-a_0)(x-a_1)\dots(x-a_{15})$.

8.54. Программа. По заданным коэффициентам многочлена 15-й степени и многочлена 8-й степени определить коэффициенты произведения этих многочленов.

8.55. Программа. По заданным коэффициентам многочлена $P(x)$ 10-й степени и многочлена $Q(x)$ 6-й степени определить коэффициенты многочлена $P(Q(x))$.

9. РЕГУЛЯРНЫЕ ТИПЫ; МАТРИЦЫ

9.1*. Однаковы ли типы `array [1..15,0..3] of char` и `array [1..15] of array [0..3] of char`?

9.2. type строка=`array [1..20] of char;`

матр1=`array [1..10] of строка;`

матр2=`array [1..15,1..20] of char;`

`var A:матр1; B:матр2;`

Указать типы переменных A , $A[3]$, $A[9][18]$, $A[1,1]$, B , $B[15]$, $B[3,3]$ и $B[10][16]$.

9.3*. `var A,B:array [1..15,0..8] of real; t:boolean;`

Какие из указанных операций допустимы в Паскале?

а) $A := B$; б) $A := A + B$; в) $t := A < > B$;

г) `read(A);` д) $A[1]:=A[15]$; е) $A[2,3]:=B[4][8]+B[1,1]$

9.4*. Программа. Ввести квадратную вещественную матрицу 4-го порядка, элементы которой заданы для ввода построчно, и распечатать ее по столбцам.

9.5*. `const n=20;`

`var A,B,C:array [1..n,1..n] of real;`
`x,y:array [1..n] of real;`

Вычислить:

а) $C = A + B$; б) $y = Ax$; в) $C = A \cdot B$;

г) $B = B^T$ (транспонировать).

9.6*. `var C,D:array [0..9,-5..3] of integer; t:boolean;`

Перемениой t присвоить значение *true*, если массивы C и D равны, и значение *false* иначе.

9.7. Программа. Даы натуральное n и (построчно) элементы квадратной вещественной матрицы A 5-го порядка. Вычислить n -ю степень этой матрицы ($A^1=A$, $A^2=A \cdot A$, $A^3=A^2 \cdot A$ и т. д.).

9.8*. type `vector=array [1..10] of real;`

`table1=array [1..8] of vector;`

`table2=array [1..8,1..10] of real;`

`var x:vector; A:table1; B:table2;`

Какие из указанных ниже операторов присваивания неверны?

а) $A := B$; б) $A[1] := A[8]$; в) $B[3] := A[3]$;

г) $B[1] := B[8]$; д) $x := A[5]$; е) $B[5] := x$

9.9. type вектор = array [1..20] of integer;
матрица = array [1..20] of вектор;

var A:матрица; x:вектор;

B:array [1..20,1..20] of integer;

Выписать фрагмент программы для решения следующей задачи:

а)* нечетные строки матрицы A заменить на x ;

б)* четные столбцы матрицы A заменить на x ;

в)* первые шесть строк массива B заменить на x ;

г) в матрице A поменять местами 1-ю и 2-ю строки,
3-ю и 4-ю строки, ..., 19-ю и 20-ю строки (воспользоваться x как вспомогательным массивом);

д) аналогичным образом поменять местами строки в B .

9.10. Программа. Данна (построчно) вещественная матрица размером 7×4 . Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент (один из них) оказался в верхнем левом углу.

9.11*. type точка = array [(x,y)] of real;

var M:array [1..40] of точка; d:real;

Рассматривая элементы массива M как координаты точек на плоскости, найти d —наибольшее расстояние между этими точками.

9.12. type страна = (Алжир, Египет, Заир, Камерун,
Конго, Ливия, Мали, Нигер,
Судан, Чад, Эфиопия);

var соседи: array [страна, страна] of boolean;
с:страна;

Элемент $соседи[a,b]$ равен *true*, если страны a и b имеют общую границу, и равен *false* иначе. Определить c —страну, имеющую наибольшее число соседей среди перечисленных стран.

9.13*. type остров = (Барбадос, Гаити, Гренада, Куба,
Мартиника, Ямайка);

месяц = (янв, фев, мар, апр, май, июн, июл, авг,
сен, окт, ноя, дек);

var t:array [остров, месяц] of real;
i:остров; m:месяц;

Элемент $t[x,y]$ означает среднемесячную температуру на острове x в месяце y . Определить, какой месяц (m) и на каком острове (i) самый холодный.

9.14*. var A:array [1..9,1..9] of real; s:real;

Найти s — сумму элементов из заштрихованной области массива A (рис. 3).

9.15. var D:array [1..10,1..10] of real; s:real;

Вычислить $s = \sum_{k=1}^{10} \max_{1 \leq i, j \leq k} D_{ij}$.

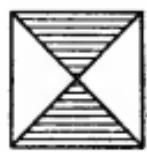
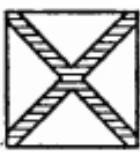
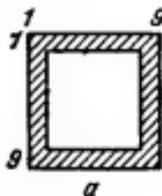


Рис. 3

9.16. var A:array [1..10,1..10] of integer;
Заполнить массив A следующим образом:

a)* $\begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 2 & \dots & 0 \\ \dots & & & & \\ 0 & 0 & 0 & \dots & 9 \end{pmatrix}$

б) $\begin{pmatrix} 1 & 2 & \dots & 10 \\ 11 & 12 & \dots & 20 \\ 21 & 22 & \dots & 30 \\ \dots & & & \\ 91 & 92 & \dots & 100 \end{pmatrix}$

в) $\begin{pmatrix} 1 & 2 & 3 & \dots & 10 \\ 0 & 1 & 2 & \dots & 9 \\ 0 & 0 & 1 & \dots & 8 \\ \dots & & & & \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}$

9.17. var A:array [1..6,1..9] of real;
x:array [1..9] of real;

Заполнить массив A по следующему правилу: $A_{ij}=x^i_j$.

9.18. type месяц=(янв, фев, мар, апр, май, июн, июл,
авг, сен, окт, ноя, дек);

дн=(пн, вт, ср, чт, пт, сб, вс, нет);

календарь=array [месяц,1..31] of дн;

var К:календарь;

Заполнить календарь K соответствующими днями недели (для несуществующих дат указать *нет*) при условии, что год невисокосный и 1 января — понедельник ($K[\text{янв},1]:=\text{пн}; K[\text{янв},2]:=\text{вт}; \dots; K[\text{фев},29]:=\text{нет}; \dots$).

9.19. var A:packed array [1..20,1..20] of boolean;
B:packed array [1..19,1..19] of boolean;
n,k:1..20;

Получить массив B из массива A удалением n -й строки и k -го столбца.

9.20. var A:array [1..15,1..20] of integer;
b:array [1..15] of boolean;

По массиву A получить массив b , присвоив его k -му элементу значение *true*, если выполнено указанное ниже условие, и значение *false* иначе:

- a)* все элементы k -го столбца массива A нулевые;
- б) элементы k -й строки массива A упорядочены по убыванию;

в) k -я строка массива A симметрична.

9.21. const n=8; m=12;

var k:integer;

C:array [1..n,1..m] of integer;

Определить k —количество «особых» элементов массива C , считая элемент «особым», если:

- а) он больше суммы остальных элементов своего столбца;
- б) в его строке слева от него находятся элементы, меньшие его, а справа—большие.

9.22. var k:integer;

C:array [1..13,1..18] of char;

Определить k —количество различных элементов массива C (т.е. повторяющиеся элементы считать один раз).

9.23. Программа. Дано вещественная матрица размером 20×30 . Упорядочить ее строки по неубыванию:

а)* их первых элементов;

б) суммы их элементов;

в) их наибольших элементов.

9.24. Имеется таблица T результатов некоторого шахматного турнира, в котором участвовало n шахматистов ($n > 2$):

T:array [1..n,1..n] of (B,H,P,X),

где $T[i,j]=B$, если i -й участник выиграл у j -го (при этом $T[j,i]=P$), $T[i,j]=H$, если i -й и j -й участники сыграли вничью, и $T[i,i]=X$. Возможный вид таблицы (при $n=3$):

(

X	B	P
P	X	H
B	H	X

)

За выигрыш дается 1 очко, за ничью—0.5 очка, за проигрыш—0 очков. Выдать на печать номера участников в порядке невозрастания набранных ими очков.

9.25. type имя=(Анна, ..., Юрий);

родство=(сын,дочь,отец,мать,нет);

var TP:array [имя,имя] of родство;
И,В,Д:имя; k:integer;

На основе (непротиворечивой) таблицы родства TP ($TP[x,y]=\text{нет}$, если y не является ни родителем, ни ребенком x 'а, $TP[x,y]=\text{сын}$, если y —сын x 'а, и т. п.) присвоить переменной:

а)* B —имя одной (любой) из внучек человека с именем I , если такие есть;

б) D —имя одного (любого) из дядей человека с именем I , если такие есть (дядями считать только братьев родителей);

в) k —количество всех кузин и кузенов (двоюродных сестер и братьев) человека с именем I .

9.26. const n=256;

type screen=packed array [1..n, 1..n] of 0..1;
var S:screen;

Преобразовать массив S , осуществив поворот элементов вокруг его центра на 90° против часовой стрелки.

9.27. Программа. Определить, является ли заданная целая квадратная матрица 10-го порядка симметричной (относительно главной диагонали).

9.28. Программа. Элемент матрицы назовем седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Для заданной целой матрицы размером 10×15 напечатать индексы всех ее седловых точек.

9.29. Программа. Данна вещественная матрица размером 7×7 , все элементы которой различны. Найти скалярное произведение строки, в которой находится наибольший элемент матрицы, на столбец с наименьшим элементом.

9.30. Программа. Определить, является ли заданная целая квадратная матрица 10-го порядка ортонормированной, т. е. такой, в которой скалярное произведение каждой пары различных строк равно 0, а скалярное произведение каждой строки на себя равно 1.

9.31. Программа. Определить, является ли заданная целая квадратная матрица 9-го порядка магическим квадратом, т. е. такой, в которой суммы элементов во всех строках и столбцах одинаковы.

9.32. Программа. Данна непустая последовательность слов из строчных латинских букв; слова разделяются запятыми, за последним словом — точка. Среди всех пар a_i

и b_i , где a_i —первая, а b_i —последняя буквы i -го слова последовательности, определить наиболее часто встречающуюся пару.

9.33. Программа. По заданным коэффициентам $a_{11}, a_{12}, \dots, a_{1n}, a_{21}, a_{22}, a_{23}, \dots, a_{2n}, \dots, a_{nn}$ ($n=20, a_{ii} \neq 0$) и правым частям b_1, b_2, \dots, b_n найти решение «треугольной» системы линейных уравнений:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ a_{33}x_3 + \dots + a_{3n}x_n = b_3 \\ \dots \\ a_{nn}x_n = b_n \end{array} \right.$$

9.34. Программа. По заданным коэффициентам A_{ij} и правым частям b_i решить систему линейных уравнений

$$\sum_{j=1}^n A_{ij}x_j = b_i \quad (i=1, 2, \dots, n; n=10),$$

считая, что ее определитель отличен от 0. (Рекомендация: систему предварительно привести к «треугольному» виду.)

9.35. Программа. Даны координаты n векторов n -мерного линейного пространства ($n=7$). Определить, являются ли они линейно независимыми.

9.36. Программа. Данна квадратная матрица n -го порядка ($n=6$). Найти матрицу, обратную ей, или установить, что такой не существует. (Замечание: если линейными преобразованиями строк привести заданную матрицу к единичной, то этими же преобразованиями единичная матрица будет приведена к искомой обратной матрице.)

10. РЕГУЛЯРНЫЕ ТИПЫ: СТРОКИ

10.1*. Какие из следующих типов являются строковыми, а какие нет и почему?

```
const n=20;
type a=array [1..30] of char;
      b=packed array [1..n] of char;
      c=packed array [0..n] of char;
      d=packed array ['1'..'9'] of char;
      e=packed array [1..n, 1..n] of char;
      f=packed array [1..10] of 'a'..'z';
      g=packed array [1..1] of char;
```

```
10.2*. const n=5;
      var a1,a2:packed array [1..n] of char;
          b1,b2:array [1..n] of char;
          c:packed array [1..5] of char;
```

Какие из указанных ниже операций допустимы в Паскале?

- а) $a1 := a2$; б) $b2 := b1$; в) $a1 := b1$; г) $c := a2$;
- д) $a2 := 'b1 + b2'$; е) $b1 := '12345'$;
- ж) $c := 'abc'$; з) $a1 < a2$; и) $b1 = b2$;
- к) $a2 > c$; л) $c > = '12345'$; м) $a1 < > '**'$;
- и) $read(c)$; о) $writeln(a1, '=', a2)$; п) $writeln(b1)$

10.3*. Программа. Данна последовательность из 60 литер. Напечатать эту последовательность дважды, в двух строках.

```
10.4*. type набор=array [1..8] of char;
        слово=packed array [1..8] of char;
        var a1:array [1..20] of набор;
            a2:array [1..30] of слово;
            k1, k2:integer;
```

Подсчитать $k1$ — количество наборов массива $a1$, совпадающих с его первым набором, и $k2$ — количество слов массива $a2$, совпадающих с его первым словом.

```
10.5.*type слово=packed array [1..5] of char;
        список=array [1..60] of слово;
        var С:список;
```

Напечатать:

- а)* все слова из списка C , отличные от слова *hello*;
- б) то слово из списка C , которое лексикографически (по алфавиту) предшествует всем другим словам этого списка (считать, что все слова различны);
- в)* текст, составленный из последних литер всех слов списка C ;
- г) все слова из списка C , содержащие ровно две буквы d .

```
10.6*. var s:packed array [1..6] of char;
```

Во входном файле *input* задано от 1 до 6 букв, за которыми следует точка. Ввести эти буквы и записать их в начало строки s , дополнив конец этой строки пробелами.

10.7. Программа. Даны два различных слова, в каждом из которых от 1 до 8 строчных латинских букв и за каждым из которых — пробел. Напечатать эти слова в алфавитном порядке. (Считать, что литера «пробел» предшествует любой литере-букве.)

```
10.8*. type color=(red,blue,green,yellow,black,white);
        var x:color;
```

Во входном файле задана последовательность строчных латинских букв, за которой следует пробел. Если это правильная запись значения типа *color*, то присвоить его переменной *x*, иначе сообщить об ошибке.

10.9. Найти ошибки в следующей программе:

```
program errors (input, output);
const sign='+-*/';
var s:packed array [1..10] of char;
    i,j,k:integer;
begin read(s);
    for i:=1 to 10 do
        for j:=1 to 4 do
            if s[i]=sign[j] then k:=k+1;
            if k<6 then writeln(s) else writeln(sign)
end.
```

10.10*. const v='aeiou';
type строка=packed array [1..200] of char;
var s:строка; k:0..200;

Определить, сколько раз (*k*) в строку *s* входят литеры, перечисленные в константе *v*.

10.11*. Программа. Дан текст из 60 литер. Напечатать только строчные русские буквы, входящие в этот текст.

10.12. Программа. Дан текст из строчных русских букв, за которым следует точка. Напечатать этот текст заглавными русскими буквами.

10.13. Программа. Дан непустой текст из заглавных русских букв, за которым следует точка. Определить, упорядочены ли эти буквы по алфавиту.

10.14. Программа. Напечатать в алфавитном порядке все различные строчные русские буквы, входящие в заданный текст из 200 литер.

10.15. type строка=packed array [1..80] of char;
var s:строка;

Известно, что в начале строки *s* находится не более 40 латинских букв, за которыми следуют пробелы. Напечатать эту строку, предварительно преобразовав ее следующим образом:

- а) все вхождения abc заменить на def;
- б)* удалить первое вхождение w, если такое есть (образовавшуюся «дыру» заполнить последующими буквами, а в конец добавить пробел);
- в) удалить все вхождения th;
- г)* заменить на ks первое вхождение x, если такое есть;

д) после каждой буквы с добавить букву и;
е) заменить все вхождения ph на f, а все вхождения ed на ing.

10.16. Программа. Даана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами — занятая, за последним словом — точка. Напечатать:

а) эту же последовательность слов, но в обратном порядке;

б) те слова, перед которыми в последовательности находятся только меньшие (по алфавиту) слова, а за ними — только большие;

в) эту же последовательность слов, но удалив из нее повторные вхождения слов;

г) все слова, которые встречаются в последовательности по одному разу;

д) все различные слова, указав для каждого из них число его вхождений в последовательность;

е) все слова в алфавитном порядке.

10.17. Программа. Даана последовательность, содержащая от 2 до 50 слов, в каждом из которых от 1 до 8 строчных латинских букв; между соседними словами — не менее одного пробела, за последним словом — точка. Напечатать те слова последовательности, которые отличны от последнего слова и удовлетворяют следующему свойству:

а) слово симметрично;

б) первая буква слова входит в него еще раз;

в) буквы слова упорядочены по алфавиту;

г) слово совпадает с начальным отрезком латинского алфавита (a, ab, abc и т. д.); учесть, что в диапазоне 'a'..'z' могут быть буквы, отличные от латинских букв;

д) слово совпадает с конечным отрезком латинского алфавита (z, uz, xuz и т. д.);

е) длина слова максимальна;

ж) в слове нет повторяющихся букв;

з) каждая буква входит в слово не менее двух раз;

и) в слове гласные буквы (а, е, и, о, у) чередуются с согласными.

10.18. Программа. Даана последовательность, содержащая от 2 до 30 слов, в каждом из которых от 2 до 10 латинских букв; между соседними словами — не менее одного пробела, за последним словом — точка. Напечатать все слова, отличные от последнего слова, предварительно преобразовав каждое из них по следующему правилу:

- а) перенести первую букву в конец слова;
- б) перенести последнюю букву в начало слова;
- в) удалить из слова первую букву;
- г) удалить из слова последнюю букву;
- д) удалить из слова все последующие вхождения первой буквы;
- е) удалить из слова все предыдущие вхождения последней буквы;
- ж) оставить в слове только первые вхождения каждой буквы;
- з) если слово нечетной длины, то удалить его среднюю букву.

10.19. Программа. Заданное целое число от 1 до 1999 напечатать римскими цифрами.

10.20. Программа. Дан текст из заглавных латинских букв, за которым следует пробел. Определить, является ли этот текст правильной записью римскими цифрами целого числа от 1 до 999, и, если является, напечатать это число арабскими цифрами (в десятичной системе).

10.21. Программа. Даы две буквы — латинская буква (от a до h) и цифра (от 1 до 8), например a2 или g5. Рассматривая их как координаты поля шахматной доски, на котором находится ферзь, нарисовать шахматную доску, пометив крестиками все поля, которые «бьет» этот ферзь, и ноликами все остальные поля.

10.22. Решить предыдущую задачу для шахматного коня.

10.23. Известно, что астрологи делят год на 12 периодов и каждому из них ставят в соответствие один из знаков Зодиака:

20.1—18.2	— Водолей	23.7—22.8	— Лев
19.2—20.3	— Рыбы	23.8—22.9	— Дева
21.3—19.4	— Овен	23.9—22.10	— Весы
20.4—20.5	— Телец	23.10—22.11	— Скорпион
21.5—21.6	— Близнецы	23.11—21.12	— Стрелец
22.6—22.7	— Рак	22.12—19.1	— Козерог

Написать программу, которая вводит дату некоторого дня года и печатает название соответствующего знака Зодиака.

10.24. Программа. Дан текст из 60 литер. Напечатать этот текст, подчеркивая (ставя минусы в соответствующих позициях следующей строки) все входящие в него заглавные и строчные русские буквы.

10.25. Программа. Дана последовательность, содержащая от 1 до 90 слов, в каждом из которых от 1 до 10 строчных русских букв; между соседними словами — не менее одного пробела, за последним словом — точка. Напечатать эти слова по алфавиту.

10.26. Программа. Напечатать таблицу умножения в шестнадцатеричной системе счисления.

11. ФУНКЦИИ И ПРОЦЕДУРЫ

11.1. var x, y, z:real;

Вычислить $z = (\text{sign } x + \text{sign } y) \cdot \text{sign}(x+y)$, где

$$\text{sign } a = \begin{cases} -1 & \text{при } a < 0, \\ 0 & \text{при } a = 0, \\ 1 & \text{при } a > 0. \end{cases}$$

При решении этой задачи:

а) не использовать функцию sign;

б)* определить и использовать функцию sign.

11.2*. Программа. По вещественному x вычислить величину

$$\text{sh}(x) \cdot \text{tg}(x+1) - \text{tg}^3(2 + \text{sh}(x-1)).$$

11.3*. type страна=(Австрия, Гана, Италия, Перу, США, Швеция);

континент=(Америка, Африка, Европа);

var x,y:страна; t:boolean;

Описать функцию конт(s), определяющую континент, на котором находится страна s , и использовать ее для изменения значения переменной t на противоположное, если страны x и y находятся на разных континентах.

11.4*. type натур=1..maxint;

var a,b:real; k:натур;

Описать функцию степень(x, n) от вещественного x и натурального n , вычисляющую (через умножение) величину x^n , и использовать ее для вычисления $b = 2.7^k + (a+1)^{-k}$.

11.5. Программа. Даны три натуральных числа. Определить их наибольший общий делитель.

11.6. type натур=1..maxint;

var m,n:натур;

function НОД(a,b :натур):натур;

begin while $a <> b$ do

if $a > b$ then $a := a - b$

else $b := b - a$;

НОД:=а
end;

а)* Определить, что будет выдано на печать:
m:=8; n:=6; writeln(НОД(m,n), m,n);

б) Объяснить, почему при вычислении $\text{НОД}(m, n)$ не меняются значения переменных m и n , хотя в теле функции оба параметра изменяют свое значение.

11.7. Указать ошибки в описании функции:

а) function f(a:'a'..'z'):integer;
begin f:=ord(a)-ord('p');
if f<0 then f:=-f end;
б) function g(k:integer):0..maxint;
var i,s:0..maxint;
begin s:=0;
for i:=-1 to k do s:=s+sqrt(i) end;
в) function h(x:integer):integer;
begin h(x):=(sqrt(x)+x)/2 end

11.8*. Программа. Даны отрезки a , b , c и d . Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь данного треугольника. (Определить процедуру $\text{печплощ}(x, y, z)$, печатающую площадь треугольника со сторонами x , y и z , если такой треугольник существует.)

11.9. var c,d:integer;
procedure P(x,y:integer);
begin y:=x+1 end;
procedure Q(x:integer; var y:integer);
begin y:=x+1 end;
procedure R(var x,y:integer);
begin y:=x+1 end;

а) Для каждой из этих процедур указать, какие из ее параметров являются параметрами-значениями, а какие — параметрами-переменными.

б)* Определить, что будет выдано на печать:

c:=2; d:=0; P(sqrt(c)+c,d); writeln(d);
c:=2; d:=0; Q(sqrt(c)+c,d); writeln(d);

Почему при изменении в процедуре параметра-значения соответствующий фактический параметр не меняет своего значения? Что надо сделать, чтобы он менял значение?

в)* Допустимы ли обращения $R(\sqrt{c}+c, d)$ и $R(c, d)$? Почему невыгодно объявлять параметр, не меняющийся в процедуре, параметром-переменной?

11.10. Пусть процедура $sokr(a,b,p,q)$ от целых параметров ($b \neq 0$) приводит дробь a/b к иесократимому виду p/q .

а) Какие из параметров этой процедуры обозначают исходные данные для нее, а какие—результаты? Какие параметры следует объявить параметрами-значениями, а какие—параметрами-переменными?

б)* Допустимы ли обращения $sokr(k+1,14,n,7)$ и $sokr(k,sqrt(36),k,n)$, где k и n —целые переменные?

в)* Описать данную процедуру и использовать ее для приведения дроби $1+1/2+1/3+\dots+1/20$ к иесократимому виду c/d .

11.11. Пусть процедура $maxmin(x,y)$ присваивает параметру x большее из вещественных чисел x и y , а параметру y —меньшее.

а) Какие из параметров этой процедуры обозначают исходные данные для нее, а какие—результаты? Какие параметры следует объявить параметрами-значениями, а какие—параметрами-переменными?

б)* Допустимы ли обращения $maxmin(5.2,sin(z))$ и $maxmin(z,k)$, где z —вещественная переменная, а k —целая?

в)* Описать данную процедуру и использовать ее для перераспределения значений вещественных переменных a , b и c так, чтобы стало $a \geq b \geq c$.

11.12. const n=1000;

type вектор=array [1..n] of real;

var a,b,c,d:вектор;

Пусть процедура $sum(x,y,z)$ присваивает вектору z сумму векторов x и y .

а) Объяснить, почему параметры x и y , хотя они и не обозначают результаты работы этой процедуры, невыгодно объявлять параметрами-значениями.

б)* Описать данную процедуру и использовать ее для вычисления $d=a+b+c$.

11.13. Для каждой из следующих задач ответить на вопросы:

Какую подзадачу приходится решать в ией несколько раз? Как следует описать решение этой подзадачи—как функцию или как процедуру? Сколько у этой функции или процедуры параметров? Каков их смысл? Какие из них—параметры-значения, а какие—параметры-переменные?

Описать данную функцию или процедуру и с ее помощью решить задачу.

a) type строка=packed array [1..60] of char;
var s,t:строка, k:integer;

Если в первой половине строки s менее 12 цифр и если в последней четверти строки t нет литер от 'a' до 'z', тогда вычислить k —количество литер '*', входящих в среднюю треть строки s .

б) var a,b:real; t:boolean;

Переменной t присвоить значение *true*, если уравнения $x^3+6.2x+a^2=0$ и $x^2+ax+b-1=0$ имеют вещественные корни и при этом оба корня первого уравнения лежат между корнями второго, и присвоить значение *false* во всех остальных случаях.

в) const n=10;

type матрица=array [1..n,1..n] of real;

вектор=array [1..n] of real;

var A,B,C:матрица; x,y,z,u:вектор;

Вычислить $u = Ax + By - Cz + Bx$.

г) type вектор=array [1..40] of integer;

var x,y,z:вектор;

Если наибольший элемент вектора x равен 10 и находится в первой половине этого вектора и если в векторе y нет положительных элементов, тогда все элементы вектора z , предшествующие его наибольшему элементу, заменить на их кубы. (Считать, что в каждом векторе наибольший элемент один.)

11.14. Найти ошибки в следующей программе:

```
program errors (input,output);
const n=40;
type vector=array [1..n] of real;
var a,b:vector;
function sum(var x,y:vector):vector;
  var i:integer; z:vector;
  begin for i:=1 to n do z[i]:=x[i]+y[i];
  sum:=z end;
procedure reverse(x:vector);
  var i:integer; r:real;
  begin for i:=1 to n div 2 do
    begin r:=x[i]; x[i]:=x[n+1-i];
    x[n+1-i]:=r end
  end;
begin read(a); b:=sum(a,reverse(a));
writeln(b) end.
```

11.15. Программа. Найти сумму кубов всех корней уравнения

$ex^3 - \pi x^2 - (2e+1)x + 2\pi = 0$
 $(\pi=3.1415927, e=2.7182818)$, вычисленных с точностью 0.0001.

11.16. Программа. Даны длины a, b и c сторон некоторого треугольника. Найти медианы треугольника, сторонами которого являются медианы исходного треугольника. (Замечание: длина медианы, проведенной к стороне a , равна $0.5\sqrt{2b^2+2c^2-a^2}$.)

11.17. Программа. Даны три слова, в каждом из которых от 1 до 6 строчных латинских букв и за каждым из которых следует пробел. Напечатать эти слова в алфавитном порядке.

11.18*. Программа. По заданным вещественным числам $a_0, a_1, \dots, a_{30}, b_0, b_1, \dots, b_{30}, c_0, c_1, \dots, c_{30}, x, y, z$ вычислить величину

$$\frac{(a_0x^{30} + a_1x^{29} + \dots + a_{30})^2 - (b_0y^{30} + b_1y^{29} + \dots + b_{30})^2}{c_0(x+z)^{30} + c_1(x+z)^{29} + \dots + c_{30}}.$$

11.19. Программа. По заданным 20-элементным целым массивам x и y вычислить

$$u = \begin{cases} \sum_{i=1}^{20} x_i^2 & \text{при } \sum_{i=1}^{15} x_i y_i > 0, \\ \sum_{i=10}^{20} y_i^2 & \text{иначе.} \end{cases}$$

11.20. Программа. По заданным 50-элементным вещественным массивам a, b и c вычислить

$$t = \begin{cases} \frac{\min(b_i)}{\max(a_i)} + \frac{\max(c_i)}{\min(b_i+c_i)} & \text{при } \min(a_i) < \max(b_i), \\ \max(b_i+c_i) + \min(c_i) & \text{иначе.} \end{cases}$$

11.21. Программа. Даны 30-элементные вещественные векторы x, y и z . Вычислить величину $(a, a) - (b, c)$, где a обозначает тот из этих векторов, в котором самый большой минимальный элемент (считать, что такой вектор единственный), b и c обозначают два других вектора, а (p, q) — скалярное произведение p и q .

11.22. Программа. Даны две квадратные вещественные матрицы 10-го порядка. Напечатать квадрат той из них, в которой наименьший след (сумма диагональных элементов), считая, что такая матрица одна.

11.23. Найти ошибки в следующей программе:

a) program errors (output);
const a='1234';
type string=packed array [1..4] of char;
var b:string;
procedure P(s1:string; var s2:string);
begin writeln(s1,s2) end;
begin P('abcd','efgh'); P(a,a);
b:=a; P(b,b) end.

b) program error (output);
type string=packed array [1..3] of char;
var x:string;
procedure Q(c:char; var d:char);
begin d:=succ(c) end;
begin x:='295'; Q(x[1],x[3]); writeln(x) end.

11.24*. Определить, что будет выдано на печать:

```
program print (output);
type vect=array [1..2] of real;
var a:vect; i:integer;
procedure R(var k:integer; var x:real);
begin k:=2; x:=0 end;
begin a[1]:=1; a[2]:=2;
i:=1; R(i,a[i]);
writeln(a[1],a[2])
end.
```

11.25*. type сдвиг=1..19;
шкала=packed array [1..20] of
boolean;

Описать процедуру *сдв(s,k)*, которая преобразует шкалу *s*, циклически сдвигая ее элементы на *k* позиций влево, где *k* — параметр типа *сдвиг*.

11.26. const n=15; m=20;
type матрица=агату [1..n, 1..m] of real;

Описать функцию *сум(A)*, вычисляющую величину

$$x_1x_n + x_2x_{n-1} + \dots + x_nx_1,$$

где *x_i* — максимальный элемент *i*-й строки матрицы *A*.

11.27*. type неотриц=0..maxint;

Описать функцию *F(m,n)=n!·m!/(n+m)!*, где *n* и *m* — неотрицательные целые числа. (Определить внутреннюю функцию, вычисляющую факториал.)

11.28*. type вектор=аггай [1..20] of char;

Описать процедуру *преобр*(*x,y,a,b*) от четырех векторов, которая преобразует векторы *x* и *y* к следующему виду:

$$x = (a_1, a_2, \dots, a_8, x_9, x_{10}, \dots, x_{20}),$$

$$y = (y_1, \dots, y_8, b_1, \dots, b_6, y_{12}, \dots, y_{18}, a_1, \dots, a_4).$$

11.29. type слово=packed аггай [1..9] of char;

Описать логическую функцию *перестановка*(*x,y*), проверяющую, можно ли, переставив буквы слова *x*, получить слово *y*.

11.30. const n=8; m=13;

type матрица=аггай [1..n,1..m] of real;

Описать процедуру *swap*(*A,B*), меняющую местами максимальные элементы матриц *A* и *B*. (Считать, что в каждой матрице только один максимальный элемент.)

11.31. const п=...; {целая константа > 1}

type число=аггай [1..п] of '0'..'9';

массив=аггай [1..40] of число;

Описать процедуру *sort*(*x*), упорядочивающую по неубыванию числа массива *x* следующим методом: все числа из *x* упорядочить по последней цифре и перенести во вспомогательный массив *y*; затем числа из *y* упорядочить по предпоследней цифре (при равенстве этих цифр сохранять упорядоченность по последней цифре) и записать их снова в массив *x*; далее числа из *x* упорядочить по третьей от конца цифре и перенести в массив *y*; и т. д. (Учесть, что в конце концов числа должны оказаться в *x*.)

11.32*. Перечислить локальные (в том числе формальные параметры) и глобальные имена, используемые в следующей процедуре:

```
procedure P(x:vect; var y:integer);
  const z='*';
  var c:index;
begin y:=0;
  for c:=a to b do if x[c]>z then y:=y+1
end
```

11.33*. Что будет напечатано следующей программой?

a) program print (output);

var x,y:char;

procedure P(x:integer);

const y=true;

begin writeln(x,' ',y) end;

```

procedure Q;
  var x:char;
begin x:=succ(y); y:='*'; writeln(x,' ',y)
end;
begin x:='a'; y:='5';
  P(8); writeln(x,' ',y);
  Q; writeln(x,' ',y)
end.

```

- 6) program print (output);
 type string=packed array [1..5] of char;
 var i:integer; t:string;
 procedure P(var s:string);
 begin i:=1;
 while s[i]<'9' do
 begin s[i]:=succ(s[i]); i:=i+1 end
 end;
 begin i:=1; t:='12945'; P(t);
 writeln(t[i]) end.
- b) program print (output);
 var a,b,c,d:integer;
 procedure P(var b:integer; c:integer);
 var d:integer;
 begin a:=5; b:=6; c:=7; d:=8;
 writeln(a,b,c,d) end;
 begin a:=1; b:=2; c:=3; d:=4;
 P(a,b); writeln(a,b,c,d)
 end.

11.34. Найти ошибки в следующем описании процедуры:

```

procedure errors(var x:boolean);
  const char=0; case=1;
  type a=(true,false); b=('a','b');
  var c:char;
begin if x then x:=(ord(true)=char) and
           false end

```

11.35. type индекс=1..20;
 матрица=array [индекс,индекс] of real;

Описать функцию $\max(A, n, k)$, где A — матрица, а n и k — индексы ($n < k$), вычисляющую наибольший элемент заштрихованной области матрицы A (рис. 4).

11.36. type table1=array [1..10,1..10] of integer;
 table2=array [1..20,1..30] of integer;

Описать процедуру $constr(A,B,C,D)$, которая по матрицам A , B и C типа $table1$ строит следующую матрицу D типа $table21$:

$$D = \begin{pmatrix} A & B & C \\ B & N & A \end{pmatrix}$$

где N — нулевая матрица типа $table1$.

11.37. Программа. Даны непустая последовательность слов, в каждом из которых от 1 до 6 латинских букв;

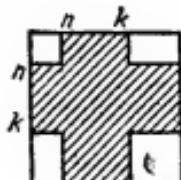


Рис. 4

между соседними словами — запятая, за последним словом — точка. Напечатать те слова, у которых одинаковые «соседи», т. е. совпадают предыдущее и следующее слова. (Определить процедуру $readword(w)$, которая вводит очередное слово и присваивает его 6-литерной строке w , а запятую или точку присваивает некоторой глобальной переменной.)

11.38. Программа. Даны 6-элементные вещественные векторы x и y и квадратные матрицы A , B и C 6-го порядка. Вычислить величину $(Ax, By) + (Cx, y)/(x, By)$.

11.39. Программа. Даны три вещественные квадратные матрицы 4-го порядка. Напечатать ту из них, норма которой наименьшая (считать, что такая матрица одна). В качестве нормы матрицы взять максимум абсолютных величин ее элементов.

11.40*. Программа. По заданным вещественным числам c и d ($c < d$) вычислить

$$\int_c^d \operatorname{arctg}^2 x dx + \int_0^\pi \sin e^{ix} dx.$$

Интегралы вычислять приближенно по формуле трапеций при $n=20$ для первого интеграла и при $n=100$ для второго:

$$\int_a^b f(x) dx \approx h \cdot \left[f(a)/2 + \sum_{i=1}^{n-1} f(a+ih) + f(b)/2 \right],$$

где $h = (b-a)/n$.

11.41. Программа. По заданным 40-элементным вещественным векторам x , y и z вычислить

$$w = \begin{cases} \prod_i (\sin(x_i) + 2) & \text{при } \prod_i (1 - y_i^2) > 0.5, \\ \prod_i (1 - z_i^2) & \text{иначе.} \end{cases}$$

11.42. Программа. Напечатать в порядке возрастания корни уравнений

$$1/(1+x^2)=x, \quad 3e^x+x=0 \text{ и } x \cdot \ln(1+x)=0.5,$$

вычисленные с заданной точностью $\epsilon > 0$.

11.43. const n=20;

type вектор=array [1..n] of real;

Описать процедуру $izm(x,y,z)$, которая в том из векторов x , y и z , где больше всего отрицательных элементов (считать, что такой вектор один), все его положительные элементы заменяет на их кубы — если это вектор x или вектор z , и на их обратные величины — если это вектор y .

11.44*. Определить, что будет выдано на печать (считать, что операнды вычисляются слева направо):

```
program sideeffect (output);
var a, b:integer;
function f(x:integer):integer;
begin f:=x; a:=0 end;
function g(var x:integer):integer;
begin g:=x; x:=0 end;
begin a:=1; write(a+f(a)); a:=-1; write(f(a)+a);
      b:=2; writeln(g(b)=g(b))
end.
```

11.45. Привести примеры, когда в Паскале не выполняются равенства $y=y$, $y+y=2*y$, b and $b=b$.

11.46*. Описать функцию $next$ без параметров, которая считывает из входного файла первую литеру, отличную от пробела, и объявляет ее своим значением. Использовать эту функцию для подсчета k — количества отличных от пробела литер текста, который задан во входном файле и за которым следует точка.

11.47.

```
const d=100; m=5;
type строка=packed array [1..d] of char;
      подстрока=packed array [1..m] of char;
      позиция=1..d;
var х:строка; у,з:подстрока;
```

Описать логическую функцию $\text{ поиск}(s, ss, k, n)$, проверяющую, входит ли подстрока ss в ту часть строки s , которая начинается с k -й позиции, и, если входит, присваивающую параметру n номер позиции, с которой начинается первое вхождение ss в эту часть строки s .

Используя данную функцию, заменить в строке x все вхождения подстроки y на подстроку z .

11.48. Программа. Даны координаты вершин двух треугольников. Определить, какой из них имеет большую площадь.

11.49. Программа. Даны координаты вершин треугольника и координаты некоторой точки внутри него. Найти расстояние от данной точки до ближайшей стороны треугольника. (При определении расстояний учесть, что площадь треугольника вычисляется и через три его стороны, и через основание и высоту.)

11.50. Программа. Три прямые на плоскости заданы уравнениями $a_kx + b_ky = c_k$ ($k=1, 2, 3$). Если эти прямые попарно пересекаются и образуют треугольник, тогда найти его площадь.

11.51. Программа. Найти наименьшее общее кратное четырех заданных натуральных чисел.

11.52. Программа. Два простых числа называются «близнецами», если они отличаются друг от друга на 2 (таковы, например, числа 41 и 43). Напечатать все пары «близнецов» из отрезка $[n, 2n]$, где n — заданное целое число, большее 2.

11.53. Программа. Два натуральных числа называются «дружественными», если каждое из них равно сумме всех делителей другого, за исключением его самого (таковы, например, числа 220 и 284). Напечатать все пары «дружественных» чисел, не превосходящих заданного натурального числа.

11.54. Программа. Даны коэффициенты многочленов $P(x)$ и $Q(x)$ 15-й степени и дано вещественное число a . Вычислить величину $P(a+Q(a)P(a+1))$.

11.55. Программа. По вещественному числу $a > 0$ вычислить величину

$$\frac{\sqrt[3]{a - \sqrt[6]{a^2 + 1}}}{1 + \sqrt[7]{3 + a}}.$$

Корни $y = \sqrt[k]{x}$ вычислять с точностью $\epsilon = 0.0001$ по следующей итерационной формуле:

$$y_0 = 1; y_{n+1} = y_n + (x/y_n^{k-1} - y_n)/k \quad (n = 0, 1, 2, \dots),$$

приняв за ответ приближение y_{n+1} , для которого $|y_{n+1} - y_n| < \varepsilon$.

11.56. Программа. По вещественным числам $\varepsilon > 0$ и t вычислить с точностью ε величину

$$\sqrt{1 - \frac{\cos^4 t}{4}} + \sqrt{1 + \frac{\arctg t}{2}} \cdot \sqrt[3]{\frac{1}{3+t^2}}.$$

Для вычисления корней использовать следующий ряд Тейлора:

$$(1+x)^a = 1 + \frac{a}{1!}x + \frac{a(a-1)}{2!}x^2 + \frac{a(a-1)(a-2)}{3!}x^3 + \dots$$

($|x| \leq 1$, $a > 0$).

11.57. Программа. Даны три целые матрицы размером 9×4 . Напечатать ту из них, где больше нулевых строк (если таких матриц несколько, напечатать их все).

11.58. Программа. Даны натуральное число p и вещественные квадратные матрицы A , B и C 4-го порядка. Получить $(ABC)^p$.

11.59. Программа. Даны вещественные матрицы A , B и C размером 10×20 . Вычислить величину

$$\frac{|A| + |B| + |C|}{|A+B+C|},$$

где $|D| = \max |D_{1,j}| + \max |D_{2,j}| + \dots + \max |D_{10,j}|$.

11.60. Программа. Даны две целые квадратные матрицы 10-го порядка. Определить, можно ли отражениями относительно главной и побочной диагоналей преобразовать одну из них в другую.

11.61. Программа. Даны вещественные числа a , b и ε ($a < b$, $\varepsilon > 0$). С точностью ε вычислить интеграл (см. упр. 5.34)

$$\int_a^b \left\{ \int_g^{f(x)} h(x,y) dy \right\} dx,$$

где $h(x,y) = e^{-y} y^{x-1}$, $g(x) = 1$,
 $f(x) = 1 + x^2$.

11.62. Программа. С заданной точностью $\varepsilon > 0$ вычислить площадь заштрихованной фигуры, показанной на рис. 5.

11.63. Программа. В точках $1, 2, \dots, k$, где k — заданное целое число от 2 до 70, напечатать (по отдельности) графики следующих функций:

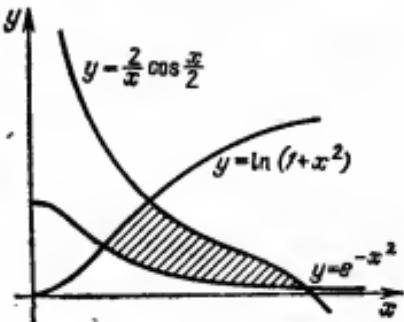


Рис. 5

$\Phi(n)$ — количество целых чисел от 1 до n , взаимно простых с числом n ;

$\tau(n)$ — количество положительных делителей числа n ;

$\pi(n)$ — количество простых чисел, не превосходящих n .

11.64. Программа. Напечатать все цифры десятичной записи чисел 2^{500} и $11+21+31+\dots+1001$. (Рекомендация: представить «длинные» натуральные числа в виде массивов из цифр и реализовать нужные операции над ними.)

11.65. Решить предыдущую задачу, но разбивая «длинные» числа не на отдельные цифры, а на группы цифр (например, по 5 соседних цифр).

11.66. Программа. Дано n вещественных чисел ($n=100$). Упорядочить их по неубыванию методом фон Неймана: завести два массива A и B и записать исходные числа в A ; упорядочить пары соседних чисел (A_1 и A_2 , A_3 и A_4 и т. д.) и записать их в B ; взять из B по две соседние упорядоченные пары и, слив их в упорядоченные четверки, снова записать в A ; затем каждые две соседние четверки из B слить в упорядоченные восьмерки и перенести в A ; и т. д.

11.67. Программа. Даны целое n от 2 до 20 и вещественное $a > 0$. Найти с точностью ε все корни n -го многочлена Чебышева $T_n(x)$, определяемого формулами

$$T_0(x) = 1; \quad T_1(x) = x; \quad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \quad (k=2, 3, \dots).$$

(Замечание: многочлен $T_k(x)$ имеет k различных корней в интервале $(-1, 1)$; если $x_1 < x_2 < \dots < x_k$ — корни многочлена $T_k(x)$, то многочлен $T_{k+1}(x)$ имеет по одному корню в каждом из интервалов $(-1, x_1), (x_1, x_2), \dots, (x_k, 1)$.)

12. РЕКУРСИЯ

12.1*.

```
function fib(n:integer):integer;
begin if n<=1 then fib:=1
      else fib:=fib(n-1)+fib(n-2)
end;
```

Вычислить $fib(2)$ и $fib(4)$.

12.2*. Какие из следующих описаний функции $f(n)$, которая должна вычислять факториал от n , правильны?

- a) function f(n:integer):integer;
begin f:=n*f(n-1) end;
- b) function f(n:integer):integer;

begin if $n=0$ then $f:=1$
else $f:=f(n+1)/(n+1)$ end;

b) function $f(n:\text{integer}): \text{integer};$
begin if $n=0$ then $f:=1$
else $f:=n*(n-1)*f(n-2)$ end;

g) function $f(n:\text{integer}): \text{integer};$
begin if $n=0$ then $f:=1$ else $f:=n*f(n-1)$ end

12.3*. Описать рекурсивную функцию $\text{pow}(x,n)$ от вещественного x ($x \neq 0$) и целого n , которая вычисляет величину x^n согласно формуле

$$x^n = \begin{cases} 1 & \text{при } n=0, \\ 1/x^{|n|} & \text{при } n<0, \\ x \cdot x^{n-1} & \text{при } n>0. \end{cases}$$

12.4*. Рекурсивно описать функцию $C(m,n)$, где $0 \leq m \leq n$, для вычисления биномиального коэффициента C_n^m по следующей формуле:

$$C_n^0 = C_n^n = 1; \quad C_n^m = C_{n-1}^m + C_{n-1}^{m-1} \text{ при } 0 < m < n.$$

12.5*. type имя=(Алла, ..., Юрий, нет);

Предполагая уже описанными функции *Отец*(x) и *Мать*(x), значениями которых являются имена соответственно отца и матери человека по имени x или идентификатор *нет*, если отсутствуют сведения о соответствующем родителе, описать логическую функцию *Потомок*(a,b), проверяющую, является ли человек с именем b потомком (ребенком, внуком, правнуком и т. п.) человека с именем a .

12.6. Решить предыдущую задачу в предположении, что имеются функция *ЧислоДетей*(x), указывающая число детей человека с именем x , и функция *Ребенок*(x,k), указывающая имя k -го ребенка человека с именем x (k не должно превышать число детей человека x).

12.7*. function $f(n:\text{integer}): \text{integer};$
begin if $n > 100$ then $f:=n-10$
else $f:=f(f(n+1))$
end;

Вычислить $f(106)$, $f(99)$ и $f(85)$. Какие вообще значения принимает эта функция?

12.8. Описать рекурсивную функцию *root*(f,a,b,eps), которая методом деления отрезка пополам находит с точностью eps корень уравнения $f(x)=0$ на отрезке $[a,b]$. (Считать, что $eps > 0$, $a < b$, $f(a) \cdot f(b) < 0$ и $f(x)$ — непрерывная и монотонная функция на отрезке $[a,b]$.)

12.9*. const n=40;

type вектор=array [1..n] of real;

Описать функцию $\min(x)$ для определения минимального элемента вектора x , введя вспомогательную рекурсивную функцию $\min1(k)$, находящую минимум среди последних элементов вектора x , начиная с k -го.

12.10. type строка=packed array [1..100] of char;

Описать рекурсивную логическую функцию *симм*(*s*, *i*, *j*), проверяющую, является ли симметричной часть строки *s*, начинаящаяся *i*-м и кончающаяся *j*-м ее элементами.

12.11*. Во входном файле задана непустая последовательность положительных вещественных чисел, за которой следует отрицательное число. Описать рекурсивную функцию *sum* без параметров для нахождения суммы этих положительных чисел.

12.12. Описать рекурсивную функцию *digits* без параметров, которая подсчитывает количество цифр в тексте, заданном во входном файле (за текстом следует точка).

12.13. Программа. Напечатать в обратном порядке заданный во входном файле текст (за текстом следует точка).

12.14. Программа. Данна последовательность ненулевых целых чисел, за которой следует 0. Напечатать сначала все отрицательные числа этой последовательности, а затем — все положительные (в любом порядке).

12.15*. Программа. Во входном файле записана (без ошибок) формула следующего вида:

Ввести эту формулу и вычислить ее значение. (Например, $5 \rightarrow 5$, $((2-4)*6) \rightarrow -12$.)

12.16. Программа. Во входном файле задан текст, за которым следует точка. Проверить, является ли этот текст правильной записью «формулы» (см. предыдущую задачу).

12.17. Программа. Во входном файле записано без ошибок логическое выражение следующего вида:

```
<логическое выражение> ::= true | false |  
                      <операция> (<операнды>)  
<операция> ::= not | and | or  
<операнды> ::= <операнд> | <операнд>, <операнды>  
<операнд> ::= <логическое выражение>
```

(У операций `and` и `or` может быть любое число operandов, у `not` — только один.)

Ввести это выражение и вычислить его значение. (Например, $\text{and}(\text{or}(\text{false}, \text{not}(\text{false})), \text{true}, \text{not}(\text{true})) \rightarrow \text{false}.$)

12.18. Программа. Во входном файле задан текст, за которым следует точка. Проверить, удовлетворяет ли его структура следующему определению:

$\langle\text{текст}\rangle ::= \langle\text{элемент}\rangle \mid \langle\text{элемент}\rangle \; \langle\text{текст}\rangle$
 $\langle\text{элемент}\rangle ::= a \mid b \mid (\langle\text{текст}\rangle) \mid [\langle\text{текст}\rangle] \mid \{\langle\text{текст}\rangle\}$

12.19. Программа. Заданный вещественный массив из n различных элементов ($n=100$) упорядочить по возрастанию следующим методом *быстрой сортировки*: выбрать какой-нибудь (например, средний) элемент массива и переставить элементы массива так, чтобы слева от выбранного элемента оказались только меньшие элементы, а справа — только большие (тем самым выбранный элемент окажется на своем окончательном месте), после чего рекурсивно применить этот же метод к левой и правой частям массива.

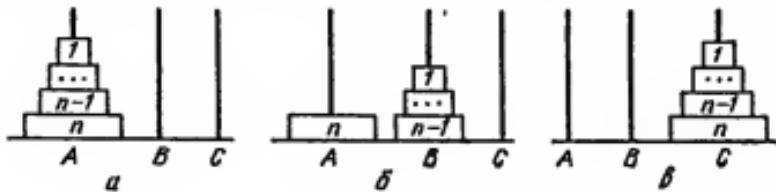


Рис. 6

12.20. («Хаинские башни»). Имеются три колышка A , B и C и n дисков разного размера, переиumerованных от 1 до n в порядке возрастания их размеров. Сначала все диски надеты на колышек A так, как показано на рис. 6, a . Требуется перенести все диски с колышка A на колышек C (рис. 6, b), соблюдая при этом следующие условия: диски можно переносить только по одному, больший диск нельзя ставить на меньший.

Написать программу, которая печатает последовательность действий (в виде «перенести диск с q на r », где q и r — это A , B или C), решающую указанную задачу для n дисков, где n — заданное натуральное число. (Подсказка! при правильном переносе n дисков с A на C обязательно встретится конфигурация, показанная на рис. 6, b .)

12.21. Программа. Имеется n населенных пунктов, переиumerованных от 1 до n ($n=10$). Некоторые пары пунктов соединены дорогами. Определить, можно ли попасть по этим дорогам из 1-го пункта в n -й.

Информация о дорогах задается в виде последовательности пар чисел i и j ($i < j$), указывающих, что i -й и j -й пункты соединены дорогой; признак конца этой последовательности — пара нулей.

12.22. Программа. Дано n различных натуральных чисел ($n=5$). Напечатать все перестановки этих чисел.

12.23. Задача о 8 ферзях: на шахматной доске расположить 8 ферзей так, чтобы они не «били» друг друга.

Написать программу, которая печатает:

- одну из таких расстановок;
- все 92 такие расстановки.

13. КОМБИНИРОВАННЫЕ ТИПЫ. ОПЕРАТОР ПРИСОЕДИНЕНИЯ

13.1. Описать комбинированный тип для представления следующего понятия:

- цена в рублях и копейках;
- * время в часах, минутах и секундах;
- дата (число, месяц, год);
- адрес (город, улица, дом, квартира);
- семинар (предмет, преподаватель, номер группы, день недели, часы занятия, аудитория);
- бланк требования на книгу (сведения о книге: шифр, автор, название; сведения о читателе: номер читательского билета, фамилия; дата заказа);
- * экзаменационная ведомость (предмет, номер группы, дата экзамена, 25 строчек с полями: фамилия студента, номер его зачетной книжки, оценка за экзамен).

13.2*. type масть=(пики, трефы, бубны, червы);

достоинство=(шесть, семь, восемь,
девять, десять,
валет, дама, король, туз);

карта=record м:масть; д:достоинство end;

Описать логическую функцию $bьет(K1, K2, KM)$, проверяющую, «бьет» ли карта $K1$ карту $K2$, с учетом того, что масть KM является козырной.

13.3*.

type строка=packed array [1..15] of char;
вершина=record название:строка;
высота:1000..9999
end;
список=array [1..30] of вершина;

Описать процедуру *СамаяВысокая(C)*, печатающую название самой высокой вершины из списка *C*.

13.4. Ответить на следующие вопросы.

а) Верно ли, что все поля записи должны быть разных типов?

б) Почему при описании записи ее поля могут перечисляться в любом порядке?

в) Верно ли, что названия полей записи могут совпадать с именами переменных, констант и других объектов программы, но не могут совпадать с названиями полей других записей?

г) Почему в переменной-поле (т. е. конструкции *r.f*) имя поля (*f*) должно указываться явно и не может быть задано в виде выражения?

13.5. Описать следующее понятие в виде массива или в виде записи, а если возможно, то в том и другом виде:

а) обозначение поля шахматной доски (a5, h8 и т. п.);

б) комплексное число;

в) точка в 50-мерном пространстве.

13.6. type точка1=аттая [(x,y)] of real;

точка2=record x,y:real end;

var p1:точка1; p2:точка2; d:real;

а) Почему допустим данный раздел типов, хотя в нем одни и теми же именами (*x* и *y*) обозначены разные объекты (индексы и поля)?

б)* Переменной *d* присвоить расстояние между точками *p1* и *p2*.

в) Допустимы ли конструкции *p1[succ(x)]* и *p2.succ(x)*?

13.7*.

type строка=packed array [1..8] of char;

адрес=record город, улица:строка;

дом, квартира:1..999

end;

var Адр1,Адр2:адрес;

Используя оператор присоединения, присвоить переменной *Адр1* значение, соответствующее адресу «Москва, ул. Арбат, д. 1, кв. 5». Кроме того, переменной *Адр2* присвоить такое же значение, заменив в нем номер дома на 17.

13.8*.

type круг=record радиус:real;

центр:record x,y:real end

end;

var К:круг;

Требуется переменной K присвоить значение, соответствующее кругу радиуса 2.5 с центром в точке (0, 1.8). В каких из следующих операторов присоединения правильно решается эта задача, а в каких нет и почему?

- a) with K do
begin радиус:=2.5; $x:=0$; $y:=1.8$ end;
- б) with K do
begin радиус:=2.5; центр. $x:=0$;
центр. $y:=1.8$ end;
- в) with K do
begin радиус:=2.5;
with центр do
begin $x:=0$; $y:=1.8$ end
end;
- г) with K , центр do
begin радиус:=2.5; $x:=0$; $y:=1.8$ end;
- д) with центр, K do
begin радиус:=2.5; $x:=0$; $y:=1.8$ end

13.9. Определить комбинированный тип для представления анкеты школьника, включающей в себя его ФИО, возраст, номера школы и класса и оценки по каким-то пяти предметам.

Описать некоторую переменную данного типа и присвоить ей значение, соответствующее следующей анкете: Петров Иван Ильич, 16 лет, 194-я школа, класс 9б, оценки 5, 3, 4, 5, 2.

13.10*. type complex=record re,im:real end;
point=record x,y:real end;
var z,w:complex; p:point; re:real;

Определить, какие значения будут иметь переменные z , w , p и re после выполнения следующих операторов:

with z do begin $re:=0$; $im:=1$ end;
 $w:=z$; $re:=2$;
with z do $re:=-1$;
with z , w do $im:=-im$;
with p do begin $x:=re$; $y:=2$ end

13.11. Найти ошибки в следующей программе:
program errors (input, output);

```
type поле=(a,b);
запись=record a:integer; b:char end;
var x,y:запись; c:char;
function f (var z:запись):запись;
var p:поле;
```

```
begin for p:=a to b do f.p:=succ(z.p) end;
begin read(c);
  with x do begin a:=ord(c); b:=c end;
  y:=x; if x=y then y:=f(x);
  with y do writeln(a,x)
end.
```

13.12. type декарт=record x, y:real end;
 поляр=record r, fi:real end;
 { $r \geq 0$, $-\pi < fi \leq \pi$ }

Описать процедуры $\Delta\pi(d, p)$, преобразующую координаты точки на плоскости из декартовых d в полярные p , и $\pi\Delta(p, d)$, выполняющую обратное преобразование.

13.13. type поле=record верт:(a,b,c,d,e,f,g,h);
 гориз:1..8
 end;

Описать логическую функцию *ходферзя*($n1, n2$), проверяющую, может ли ферзь за один ход перейти с поля $n1$ шахматной доски на поле $n2$.

13.14. type время=record час:0..23; мин, сек:0..59 end;
Описать:

а) логическую функцию *раньше* ($t1, t2$) для проверки, предшествует ли время $t1$ времени $t2$ (в рамках суток);

б)* процедуру *следсек* ($t, t1$), присваивающую параметру $t1$ времени, на 1 секунду большее времени t (учесть смену суток);

в) процедуру *интервал* ($d, t2, t1$), которая вычисляет время d , прошедшее от времени $t1$ до времени $t2$: $d=t2-t1$ (считать, что $t2>t1$).

13.15. type имя=(Аня, Валя, Женя, Петя, Саша, Таня,
 Шура, Юра);
 данные=record пол:(муж, жен);
 рост:140..200 end;
 группа=array [имя] of данные;

Описать:

а) функцию *средрост* (GP), определяющую средний рост женщин из группы GP ;

б) функцию *высокий* (GP) для определения имени самого высокого мужчины из группы GP ;

в) логическую функцию *одинрост* (GP), проверяющую, есть ли в группе GP хотя бы два человека одного роста.

13.16. type рац=record числ:integer;
 знам:1..maxint end;
 массив=array [1..20] of рац;

Описать:

а)* логическую функцию *равно(a,b)*, сравнивающую два рациональных числа *a* и *b*;

б) процедуру *слож(c,a,b)*, которая складывает рациональные числа *a* и *b* и присваивает их сумму рациональному параметру *c*;

в) процедуру *сокр(r)*, приводящую рациональное число *r* к несократному виду;

г) процедуру *max(x,m)*, присваивающую параметру *m* наибольшее из рациональных чисел массива *x*.

13.17. type complex=record re, im:real end;
coeff=record a,b,c:complex end; {a \neq 0}

Описать процедуру *value(p,x,y)*, которая вычисляет *y* — значение квадратного трехчлена $ax^2 + bx + c$ с коэффициентами из *p* в комплексной точке *x*.

13.18. type костьдомино=record лев, прав:0..6 end;
ряд=аггау [1..28] of костьдомино;

Описать логическую функцию *правильныЙ ряд(r)*, которая проверяет, правильно ли выставлены кости домино в ряду *r* (равна ли правая цифра очередной кости левой цифре следующей кости).

13.19. type число=1..31; месяц=1..12;
год=1..2000;
дата=record ч:число; м:месяц;
г:год end;
деньнедели=(пн,вт,ср,чт,пт,сб,вс);

Считая, что все даты даются по григорианскому календарю («новому стилю»), описать:

а) функцию *послчисло(d)*, вычисляющую количество дней в том месяце, которому принадлежит дата *d*;

б) логическую функцию *вернаядата(d)*, проверяющую правильность даты *d* (т. е. чтобы не было 31 июня и т. п.);

в) функцию *числодней(d)*, подсчитывающую, сколько дней прошло от 1 января 1-го года нашей эры до даты *d*;

г) функцию *ДН(d)* для определения дня недели, на который приходится дата *d* (учесть, что 1 января 1-го года нашей эры было понедельником).

13.20. type строка=packed аггау [1..20] of char;
житель=record

фамилия, город:строка;
адрес:record улица:строка;
дом,
квартира:1..999

```
        end  
    end;  
список=array [1..15] of житель;
```

Описать процедуру *ИронияСудьбы(С)*, которая печатает фамилии двух (любых) жителей из списка *С*, живущих в разных городах по одинаковому адресу.

```
13.21. type строка=packed array [1..16] of char;  
      data=record число:1..31;  
             месяц:1..12;  
             год:1900..1979  
         end;  
      анкета=record фамилия:строка;  
             пол:(муж, жен);  
             деньрожд:дата  
         end;  
      группа=array [1..25] of анкета;
```

Описать:

а) процедуру *старший(Гр, Фам)*, присваивающую строке *Фам* фамилию самого старшего мужчины из группы *Гр* (считать, что такой есть и он единственный);

б) процедуру *печ(Гр, Бук)*, печатающую все фамилии людей из группы *Гр*, начинающиеся с литер *Бук*, и даты рождения этих людей.

```
13.22. type слово=packed array [1..9] of char;  
      номер телефона=1000000..9999999;  
      знакомый=record фамилия:слово;  
                     номер:номер телефона  
                 end;  
      страница=array [1..20] of знакомый;  
      записная книжка=array ['A'..'Z'] of  
                     страница;
```

Считая, что на каждой странице записи о книжке указанны фамилии, начинающиеся с одной и той же буквы — индекса этой страницы, описать логическую функцию:

а) *номер(ЗП, Ф, НТ)*, определяющую, есть ли в записи о книжке *ЗП* сведения о знакомом с фамилией *Ф*, и, если есть, присваивающую параметру *НТ* номер его телефона;

б) *фамилия(ЗП, НТ, Ф)*, определяющую, есть ли в записи о книжке *ЗП* сведения о знакомом, имеющем телефон с номером *НТ*, и, если есть, присваивающую параметру *Ф* фамилию этого знакомого.

13.23. const n=300;
 type запись=record ключ:integer;
 тело:array [1..99] of 'a'..'z'
 end;
 таблица=array [1..n] of запись;

Считая, что в таблице записи имеют различные ключи, описать:

- а) процедуру *упор*(*T*), упорядочивающую записи таблицы *T* по возрастанию их ключей;
- б) логическую функцию *поиск*(*T,K,H*), определяющую, есть ли в таблице *T* (все записи которой уже упорядочены по возрастанию их ключей) запись с ключом *K*, и, если есть, присваивающую ее номер параметру *H*.

13.24. Промоделировать работу машины Тьюринга (см., например, [9]):

— выбрать представление для ленты, автомата и программы машины, считая, что лента имеет конечный размер, который так же, как и алфавит машины и набор состояний автомата, известен заранее;

— написать процедуру, которая начиная с исходной конфигурации (автомат находится в определенном состоянии и «видит» самый левый символ входного слова) моделирует работу машины согласно заданной программе. (Если машина не останавливается через определенное число шагов, процедура должна прекратить вычисления.)

13.25. Программа. Даны комплексное число *z* (пара вещественных чисел) и вещественное число *e* > 0. Вычислить с точностью *в* значение следующей комплексной функции:

- а) $e^z = 1 + z/1! + z^2/2! + \dots + z^n/n! + \dots$;
- б) $\operatorname{sh} z = z + z^3/3! + z^5/5! + \dots + z^{2n+1}/(2n+1)! + \dots$;
- в) $\operatorname{ch} z = 1 + z^2/2! + z^4/4! + \dots + z^{2n}/(2n) + \dots$;
- г) $\sin z = z - z^3/3! + z^5/5! - \dots + (-1)^n z^{2n+1}/(2n+1)! \dots$;
- д) $\cos z = 1 - z^2/2! + z^4/4! - \dots + (-1)^n z^{2n}/(2n)! + \dots$;
- е) $\ln(1+z) = z - z^2/2 + z^3/3 - \dots + (-1)^{n-1} z^n/n + \dots$
 $\dots(|z| < 1)$;
- ж) $\operatorname{arctg} z = z - z^3/3 + z^5/5 - \dots + (-1)^n z^{2n+1}/(2n+1) + \dots$
 $\dots(|z| < 1)$.

13.26. Программа. Найти корни квадратного трехчлена с заданными комплексными коэффициентами.

13.27. Во входном файле содержится информация об итогах зимней сессии на 1 курсе. Сведения о каждом

студенте-первокурснике (всего их 400) заданы в виде следующего текста:

〈фамилия〉, 〈номер группы〉,
〈оценка1〉, 〈оценка2〉, 〈оценка3〉

причем в фамилии—не более 12 букв, номер группы — целое от 101 до 116, каждая оценка — это 2, 3, 4 или 5, причем первая оценка — за экзамен по математике, вторая — по алгебре, третья — по программированию. Сведения о студентах отделены друг от друга точкой с запятой.

Написать программу, которая вводит эту информацию и печатает следующие данные:

- фамилии студентов, имеющих задолженность хотя бы по одному предмету;
- «качество» успеваемости, т. е. процент студентов, сдавших все экзамены на 5 и 4;
- название предмета, который был сдан лучше всего;
- номера групп в порядке убывания средней успеваемости их студентов.

13.28. Во входном файле записана следующая информация о каждом из 2000 студентов некоторого вуза:

〈фамилия〉, 〈имя〉, 〈отчество〉, 〈пол〉, 〈возраст〉, 〈курс〉
причем в фамилии, имени и отчестве — не более 12 букв, пол указывается буквами М и Ж, возраст — целое от 16 до 35, курс — целое от 1 до 5. Сведения о студентах отделены друг от друга точкой с запятой.

Написать программу, которая вводит эту информацию и печатает следующие данные:

- номер курса, на котором наибольший процент мужчин;
- самые распространенные мужское и женское имена;
- фамилии (в алфавитном порядке) и инициалы всех студенток, возраст и отчества которых являются одновременно самыми распространенными.

14. МНОЖЕСТВЕННЫЕ ТИПЫ

14.1. type bits = set of 0..1;
var x:bits; y:set of (a,b,c);
z:set of '*'..'*';

Ответить на следующие вопросы:

- Каков базовый тип каждого из указанных множественных типов?
- * Сколько и какие значения может принимать каждая из переменных *x*, *y* и *z*?

14.2*. Если в базовом типе *п* различных значений, то сколько различных значений в построенном на его основе множественном типе?

14.3*. type деньнедели=(пн,вт,ср,чт,пт,сб,вс);
Описать множественный тип, включающий в себя множества из:

- а) названий любых дней недели;
- б) названий рабочих дней недели.

14.4*. Какие из следующих описаний неверны и почему?

```
type точкн=set of real;
    байт=packed array [1..8] of 0..1;
    данные=set of байт;
    месяц=(янв, фев, мар, апр, май, июн, июл, авг,
            сен, окт, ноя, дек);
    М1=set of месяц;
    М2=set of июн..авг;
    М3=set of дек..фев;
    М4=set of (июн, июл, авг);
```

14.5*. Какие из следующих конструкций являются множествами (в смысле языка Паскаль), а какие нет и почему?

- а) [9,6,3,0]; б) [2..3,5,7];
- в) [1..15,4..18]; г) ['*', '*'];
- д) [0..0]; е) [true..false];
- ж) [2, sqrt(9)]; з) ['=', '>=', '>'];
- и) [[], [5]]; к) [odd(7), 0<2]

14.6*. var p:set of 0..9; i, j:integer;

Если $i=3$ и $j=5$, то какое значение получит переменная *p* при выполнении следующего оператора присваивания:

- а) $p:=[i+3, j \text{ div } 2, j..sqr(i)-3];$
- б) $p:=[2*i..j];$
- в) $p:=[i, j, 2*i, 2*j]$

14.7. var s:set of char; c, d:char;

Переменной *s* присвоить:

- а) пустое множество;
- б) множество из строчных гласных латинских букв (а, е, и, о, у);
- в) множество из всех цифр;
- г) множество литер, которые больше *c*, но меньше *d* ($c < d$).

14.8*. Вычислить значения отношений:

- а) $[2]<>[2,2,2];$ б) $['a','b']=['b','a'];$
- в) $[4,5,6]=[4..6];$ г) $['c','b']=['c'..'b'];$
- д) $[2,3,5,7]<=[1..9];$ е) $[3,6..8]<=[2..7,9];$

- ж) $[] \leq [0..9]$; з) 'q' in ['a'..'z'];
- и) trunc(3.9) in [1,3,5]; к) odd(4) in [];
- л) [2] < [1..3]; м) 66 = [66]

14.9. Эквивалентны ли выражения:

- а) p in [0,5,19] и (p=0)or(p=5)or(p=19)?
- б) p in [20..50] и (p>=20)and(p<=50)?

14.10*. type строка=packed array [1..100] of char;

Описать функцию *счет(s)*, подсчитывающую общее количество цифр и знаков '+', '-' и '*', входящих в строку *s*.

14.11. Программа. Даюо 100 целых чисел от 1 до 50. Определить, сколько среди них чисел Фибоначчи и сколько чисел, первая значащая цифра в десятичной записи которых — 1 или 2.

14.12. type месяц=1..12;

Описать функцию *числодней(m)*, определяющую количество дней в месяце *m* (невисокосного года).

14.13*. type M=set of 0..99;

Описать функцию *card(A)*, подсчитывающую количество элементов в множестве *A* типа *M*. (Например, *card([5, 8, 23])=3*.)

14.14*. type letters=set of 'a'..'z';

Описать процедуру *print!(A)*, печатающую в алфавитном порядке все элементы множества *A*, имеющего тип *letters*.

14.15. const n=10;

type номер=1..n;

матрица=array [номер, номер] of real;

ном=set of номер;

Описать функцию *sum(A, s1, s2)*, вычисляющую сумму тех элементов матрицы *A*, номера строк и столбцов которых принаследуют соответственно непустым множествам *s1* и *s2* типа *ном*.

14.16*. type деньнедели=(пн,вт,ср,чт,пт,сб,вс);

рабочийдень=пн..пт;

var wd:деньнедели; t:boolean;

Требуется переменной *t* присвоить значение *true*, если *wd* — рабочий день, и значение *false* иначе. Какими из следующих операторов правильно решается эта задача?

- а) *t:=wd* in рабочийдень; б) *t:=wd=рабочийдень*;
- в) *t:=wd* in [рабочийдень]; г) *t:=wd* in [пн..пт];
- д) *t:=[wd]<=[пн..пт]*; е) *t:=[wd]=[пн..пт]*

14.17*. Вычислить значения выражений:

- а) *[1,3,5]+[2,4]*; б) *[1,3,5]*[2,4]*; в) *[1,3,5]-[2,4]*;

- г) $[1..6] + [3..8]$; д) $[1..6]*[3..8]$; е) $[1..6] - [3..8]$;
 ж) $[2, 4] + [1..5]$; з) $[2, 4]*[1..5]$; и) $[2, 4] - [1..5]$;
 к) $[] + [4]$; л) $[] * [4]$; м) $[] - [4]$

14.18. Вычислить значения выражений:

- а)* $[2..13]*[3, 13..60] + [4..10] - [5..15]*[6]$;
 б) $[2..10] - [4, 6] - [2..12]*[8..15]$;
 в) $(['0'..'7']+['2'..'9'])*(['a']+['z'])$

14.19. Упростить (A и B —множества):

- а)* $A*B - A$; б)* $A - (A - B)$;
 в) $(A+B) - (A-B) - (B-A)$; г) $(A-B) + (B-A) + A*B$.

14.20. Не используя дополнительные переменные, поменять местами значения переменных-множеств A и B .

14.21*. var x,y,z:set of 8..22;

Переменной x присвоить множество всех целых чисел от 8 до 22, переменной y —множество всех простых чисел из этого диапазона, а переменной z —множество всех составных чисел из этого же диапазона.

14.22. Программа. Дан текст из цифр и строчных латинских букв, за которым следует точка. Определить, каких букв—гласных (а, е, и, о, у) или согласных—больше в этом тексте.

14.23*. var A,B:set of char; x:char;

Переменной B присвоить множество, полученное из A :

- а) добавлением элемента x ;
 б) удалением элемента x .

14.24. type натур=1..maxint;

Описать:

а)* функцию $digits(n)$, подсчитывающую количество различных (значащих) цифр в десятичной записи натурального числа n ;

б) процедуру $print(n)$, печатающую в возрастающем порядке все цифры, не входящие в десятичную запись натурального числа n .

14.25. Программа. Дан текст из строчных латинских букв, за которым следует точка. Напечатать:

а)* первые вхождения букв в текст, сохраняя их исходный взаимный порядок;

б) все буквы, входящие в текст не менее двух раз;

в) все буквы, входящие в текст по одному разу.

14.26. Программа. Дан текст, за которым следует точка. В алфавитном порядке напечатать (по разу) все строчные

русские гласные буквы (*a, e, i, o, u, ы, э, ю, я*), входящие в этот текст.

14.27. Программа. В возрастающем порядке напечатать все целые числа из диапазона 1..10 000, представимые в виде n^3+m^2 , где $n, m \geq 0$.

14.28. Программа. В порядке убывания напечатать все целые числа из диапазона 1..4900, которые представимы в виде n^2+2k^2 , но не представимы в виде $7ij+j+3$ ($i, k, l, j \geq 0$).

14.29. Программа. Дано целое n от 2 до 1000. Используя метод «решета Эратосфена», напечатать в убывающем порядке все простые числа из диапазона $n..2n$.

(Суть этого метода: выписываются все целые числа, большие 1; выбирается первое из них (это 2, простое число) и вычеркиваются все кратные ему числа, кроме него самого; затем берется следующее из невычеркнутых чисел (это 3, также простое число) и вычеркиваются все кратные ему, опять же кроме него самого; и так для каждого не вычеркнутого ранее числа. В конце концов останутся только простые числа начиная с 2.)

14.30. type продукт=(хлеб, масло, молоко, мясо,
рыба, соль, сыр, колбаса,
сахар, чай, кофе);
ассортимент=set of продукт;
магазины=аггай [1..20] of ассортимент;

Описать процедуру *Наличие(Mag, A, B, C)*, которая по информации из массива *Mag* типа *магазины* (*Mag[i]*—это множество продуктов, имеющихся в i -м магазине) присваивает параметрам *A*, *B* и *C* типа *ассортимент* следующие значения:

A—множество продуктов, которые есть во всех магазинах;

B—множество продуктов, каждый из которых есть хотя бы в одном магазине;

но *C*—множество продуктов, которых нет ни в одном магазине.

14.31. type имя=(Вася, Володя, Ира, Лиза, Марина,
Миша, Наташа, Олег, Оля, Света,
Юля);

гости=set of имя;

группа=аггай [имя] of гости;

Описать логическую функцию *Вездe(GP)*, определяющую, есть в группе *GP* хотя бы один человек, побывавший

в гостях у всех остальных из группы ($GP[x]$ — множество людей, бывших в гостях у человека с именем x ; $x \notin GP[x]$).

14.32. type город=(a,b,c,d,e,f,g,h);

города=set of город;

рейсы=array [город] of города;

Описать процедуру *МожноПопасть*(P,H,K), которая по рейсам P ($P[x]$ — множество городов, в которые можно за один рейс доехать автобусом из города x) определяет K — множество городов, в которые можно попасть автобусом (за один рейс или через другие города) из города H .

14.33. Описать логическую функцию *path*(G,N,K,D), которая определяет, есть ли в ориентированном графе G путь из вершины N в вершину K , и, если есть, присваивает параметру D длину (число дуг) кратчайшего пути из N в K .

Использовать следующее представление графа:

```
type вершина=(b1,b2,b3,b4,b5,b6,b7,b8);
соседи=set of вершина;
граф=array [вершина] of соседи;
```

($G[x]$ — множество вершин, в которые ведут дуги из вершины x .)

14.34. Найти все ошибки в следующем фрагменте программы:

```
type M=set of char;
function f(a,b:M; x:char):M;
begin if a*b=0 then a:=[x] else
      if a<b then a:=b+x else
      if ord(x) in a-b
          then a:=a-[x..!<='];
      f:=a+b
end;
```

14.35. Программа. Даны непустая последовательность слов из строчных русских букв; между соседними словами — запятая, за последним словом — точка. Напечатать в алфавитном порядке:

- а) все гласные буквы, которые входят в каждое слово;
- б) все согласные буквы, которые не входят ни в одно слово;
- в) все звонкие согласные буквы, которые входят хотя бы в одно слово;
- г) все глухие согласные буквы, которые не входят хотя бы в одно слово;

д) все согласные буквы, которые входят только в одно слово;

е) все глухие согласные буквы, которые не входят только в одно слово;

ж) все звонкие согласные буквы, которые входят более чем в одно слово;

з) все гласные буквы, которые не входят более чем в одно слово;

и) все звонкие согласные буквы, которые входят в каждое нечетное слово и не входят ни в одно четное слово;

к) все глухие согласные буквы, которые входят в каждое нечетное слово и не входят хотя бы в одно четное слово.

(Примечание: гласные буквы — а, е, и, о, у, ы, ю, я (ѣ обычно не входит в литературный тип); согласные — все остальные буквы, кроме й, ь, ъ; звонкие согласные — б, в, г, д, ж, з, л, м, н, р; глухие согласные — к, п, с, т, ф, х, ц, ч, ш, ѿ.)

14.36. Программа. Дан некоторый текст, за которым следует точка (в сам текст точка не входит). Определить, является ли этот текст правильной записью «формулы»:

```
<формула> ::= <терм> | (<формула> <знак> <формула>)
<знак> ::= + | - | *
<терм> ::= <имя> | <целое>
<имя> ::= <буква> | <имя> <буква> | <имя> <цифра>
<целое> ::= <цифра> | <целое> <цифра>
<буква> ::= а | б | в | г | д | е | ж
<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

14.37. Программа. Вычислить определитель заданной квадратной матрицы A n -го порядка ($n=15$), используя формулу разложения по первой строке:

$$\det(A) = \sum_{k=1}^n (-1)^{k+1} a_{1k} \cdot \det(A_k),$$

где A_k — матрица, полученная из A удалением 1-й строки и k -го столбца. (Рекомендация: определить рекурсивную функцию от параметров l и s , которая по указанной формуле вычисляет определитель матрицы, полученной из A удалением первых l строк и всех столбцов, номера которых принадлежат множеству s .)

15. ФАЙЛОВЫЕ ТИПЫ

15.1. Ответить на следующие вопросы.

а) Верно ли, что элементы файла должны быть одного типа и что файл отличается от массива только тем, что

размер (количество элементов) файла произволен, а размер массива фиксирован?

б) Можно ли, считав из файла пятый элемент, затем сразу же считать второй элемент? А какой можно?

в) Верно ли, что, считав из файла пятый элемент, затем уже никогда нельзя считать его второй элемент?

г) В какое место файла можно добавлять новые элементы: в начало, в середину, в конец, куда угодно, никуда?

д) Если не переписывать файл заново, то значения каких его элементов можно менять: только первого, только последнего, каких угодно, никаких? А какие элементы можно удалять из файла (при том же условии)?

е) Верно ли, что в одно и то же время нельзя считывать из файла и записывать в него? Верно ли, что, начав считывать из файла, затем уже никогда нельзя записывать в него? А наоборот?

ж) Можно ли сравнивать файлы или присваивать один файл другому?

15.2*. var f:file of integer; x,y:integer;

Пусть файл *f* содержит два элемента—3 и 7. Определить, какое значение будет иметь переменная *y* после выполнения следующих операторов:

а) reset(f); read(f,y);
if not eof(f) then read(f,y);
if not eof(f) then read(f,y);

б) reset(f); y:=0;
while not eof(f) do
begin read(f,x);
y:=y + x end;

в) reset(f); y:=1;
repeat read(f,x); y := y*x until eof(f)

15.3. type слово=file of char;

Найти ошибки в приведенном ниже описании функции *длина(w)*, которая должна определять количество элементов в произвольном слове *w*:

```
function длина (w:слово):integer;
var k:integer; c:char;
begin reset(w); k:=0;
repeat read(w, c); k:=k + 1 until eof(w);
длина:=k
end
```

15.4*. type серия = file of real;

Описать функцию *отриц(s)*, подсчитывающую сумму отрицательных элементов в серии *s*.

15.5.

type цена = record руб:0..maxint; коп:0..99 end;
прайскурант = file of цена;

Описать процедуру *min(P, Ц)*, присваивающую параметру *Ц* наименьшую цену из непустого прайскуранта *P*.

15.6*. type ряд = file of 0..999;

Описать логическую функцию *упор(r)*, проверяющую, упорядочены ли по возрастанию элементы непустого ряда *r*.

15.7*. type текст = file of char;

Описать логическую функцию *eq(t1, t2)*, проверяющую тексты *t1* и *t2* на равенство.

15.8.

type время = record час:0..23; мин, сек:0..59 end;
ФВ = file of время;

Описать логическую функцию *eq(f, g)*, проверяющую на равенство файлы *f* и *g* типа *ФВ*.

15.9. type слово = file of char;

Описать логическую функцию *less(w1, w2)*, проверяющую, предшествует ли лексикографически слово *w1* слову *w2*.

15.10. type FR = file of real;

Описать функцию *предпосл(f)*, значением которой является предпоследний элемент файла *f*, имеющего тип *FR* и содержащего не менее двух элементов.

15.11*. var f:file of integer; i:integer;

Определить содержимое файла *f* после выполнения следующих операторов:

в) rewrite(f);
if eof(f) then write(f,1) else write(f,2);
if eof(f) then write(f,3) else write(f,4);

б) rewrite(f);
for i:=3 downto 1 do write(f, sqr(i))

15.12*. type строка = packed array [1...100] of char;
текст = file of char;

Описать процедуру *цифры(s, t)*, которая записывает в текст *t* все цифры из строки *s*.

15.13. type ряд = file of 1..maxint;

Описать процедуру *fib(f,n)*, записывающую в ряд *f* все числа Фибоначчи (1, 1, 2, 3, 5, ...), не превосходящие целого положительного числа *n*.

15.14*. type FB = file of boolean;

Описать процедуру *присв(f,g)* от двух файлов типа *FB*, которая файлу *f* присваивает содержимое файла *g*.

15.15. type letters = file of 'a'..'z';

Описать процедуру *append(f,g,h)* от трех файлов типа *letters*, которая записывает в файл *f* сначала все элементы файла *g*, а затем—все элементы файла *h*.

15.16. type data = record месяц: (янв, фев, мар, апр,
май, июн, июл, авг, сен,
окт, ноя, дек);
число: 1..31

end;

ФД = file of data;

Описать процедуру *зап(d,s,w)* от трех файлов типа *ФД*, которая из файла *d* переписывает в файл *s* все летние даты, а в файл *w*—все зимние даты.

15.17*. type reals = file of real;

Описать функцию *less(f)* от непустого файла *f* типа *reals*, которая подсчитывает количество элементов файла *f*, меньших среднего арифметического всех элементов этого файла.

15.18.

type человек = record
 имя: packed array [1..9] of char;
 возраст: 1..99
end;

группа = file of человек;

Описать процедуру *СамыеМолодые(ГР)*, печатающую имена всех людей из непустой группы *ГР*, имеющих наименьший возраст.

15.19. Программа. Даны непустая последовательность слов, содержащих от 1 до 8 букв; между соседними словами—запятая, за последним словом—точка. Напечатать все слова, отличные от последнего слова.

15.20. Программа. Даны непустая последовательность слов, содержащих от 1 до 8 букв; между соседними словами—запятая, за последним словом—точка. Напечатать все слова наименьшей длины.

15.21. type текст = file of char;

Описать процедуру:

- а) *add1(t,c)*, добавляющую литеру *c* в начало текста *t*;
б) *addlast(t,c)*, добавляющую литеру *c* в конец текста *t*;

- в) *double(t)*, удваивающую в тексте *t* каждую цифру;
г) *replace(t,c)*, заменяющую последнюю литеру непустого текста *t* на литеру *c*;

- д) *next(t)*, заменяющую в тексте *t* каждую цифру на следующую по величине цифру ('9' заменять на '0');

- е) *delete(t)*, удаляющую из текста *t* все литеры '+' и '-';

- ж) *del(t)*, удаляющую из текста *t* предпоследний элемент, если такой есть;

- з) *firsts(t)*, оставляющую в тексте *t* только первые вхождения каждой литеры.

15.22*. var f:file of integer; x:integer;

Пусть файл *f* содержит элементы 1 и 2. Какое значение будет иметь переменная *x* после выполнения следующих операторов?

- а) *reset(f); if f† = 1 then get(f); x:=f†;*
б) *reset(f); x:=0;*
 *if not eof(f) then begin get(f);
 x:=x+f† end;*
 if not eof(f) then begin x:=x+f†; get(f) end;
 if not eof(f) then x:=x+f†;
в) *reset(f); get(f); get(f); x := f†;*
г) *reset(f); read(f,x); if f† > 1 then read(f,x)*

15.23*. var t:file of char; c:char;

Каким будет содержимое файла *t* после выполнения следующих операторов?

- а) *rewrite(t);
 if eof(t) then t† := 'a' else t† := 'b'; put(t);*
б) *rewrite(t); put(t); t† := '*';*
в) *rewrite(t);
 for c := '1' to '3' do
 begin t† := c; put(t) end;*
г) *rewrite(t); t† := '1';
 for c := '3' downto '1' do
 begin put(t); t† := c end*

15.24. Описать тип (любой из возможных) переменной *x* так, чтобы выражение *x†[5].y+ [5]* было правильным.

15.25*. При условии, что известен тип (Φ) файлов f и g , но не известен тип их элементов, описать процедуру $pris(f,g)$, присваивающую файлу f содержимое файла g .

15.26*. type FR=file of real;

Описать логическую функцию $mid(f,m)$, которая определяет, имеет ли файл f типа FR нечетную длину, и, если имеет, присваивает параметру m средний элемент этого файла.

15.27. type FR=file of real;

Описать функцию $incr(f)$ определяющую количество элементов в наиболее длинной возрастающей последовательности файла f .

15.28. type FI=file of integer;

Пусть в каждом из файлов f и g элементы упорядочены по неубыванию. Требуется слить эти файлы в один файл h , также упорядоченный по неубыванию.

Решение этой задачи описать в виде процедуры $merge(f,g,h)$ от параметров типа FI .

15.29. type файл=file of char;

Описать логическую функцию $relation(f,v)$, проверяющую, является ли содержимое файла f правильной записью «отношения» (см. ниже), и, если является, присваивающую логическому параметру v значение этого отношения.

```
<отношение> ::= <число> <знак отношения> <число>
<знак отношения> ::= <|=|=|> | <=|=|> |=
<число> ::= <цифра> | <цифры>
<цифры> ::= <неноль> <цифра> | <цифры> <цифра>
<неноль> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<цифра> ::= 0 | <неноль>
```

15.30. Считая t текстовым файлом (файлом типа $text$), ответить на следующие вопросы:

а) Эквивалентны ли типы $text$ и $file of char$?

б) Кроме текстовых файлов, файлы каких еще типов могут делиться на строки? Обязательно ли все строки файла должны быть одинаковой длины? Допустимы ли пустые строки?

в) Если значение $eoln(t)$ равно $true$, то чему равно значение t^{\dagger} ?

г) Если при записи в t надо закончить строку, то как это сделать? Какие действия влечет выполнение оператора $writeln(output)$?

д) Верно ли, что из текстового файла можно считы-

вать только по одной лите^р? А записывать? Если k —целая переменная, то допустимы ли операторы $\text{read}(t,k)$ и $\text{write}(t,k)$?

е) Какого типа файлы input и output ? Почему они не описываются в программах? Почему в программах не пишут $\text{reset}(\text{input})$ и $\text{rewrite}(\text{output})$? А что будет, если написать в программе эти операторы? Допустимы ли операторы $\text{write}(\text{input},5)$ и $\text{read}(\text{output},x)$, где x —вещественная переменная?

ж) Эквивалентны ли обращения $\text{read}(\text{input},x)$ и $\text{read}(x)$, $\text{eof}(\text{input})$ и eof , $\text{get}(\text{input})$ и get ?

15.31*. Описать процедуру $\text{trangle}(t)$, формирующую текстовый файл t из 9 строк, в первой из которых—одна лите^р '1', во второй—две лите^р '2', ..., в девятой—девять лите^р '9'.

15.32. Описать процедуру $\text{line40}(t)$, которая считывает из входного файла лите^р до первой точки и записывает их (без точки) в текстовый файл t , формируя в нем строки по 40 лите^р (в последней строке лите^р может быть и меньше).

15.33. Описать функцию, которая:

а)* подсчитывает количество пустых строк в текстовом файле t ;

б) находит максимальную длину строк текстового файла t .

15.34. Описать процедуру $\text{printlines}(t)$, печатающую построчно содержимое текстового файла t .

15.35. Пусть текстовый файл t разбит на непустые строки. Описать функцию $\text{count}(t)$ для подсчета числа строк, которые:

а)* начинаются с буквы d ;

б) оканчиваются буквой z ;

в) начинаются и оканчиваются одной и той же лите^рой;

г) состоят из одинаковых лите^р.

15.36*. Описать процедуру $\text{присв}(t1, t2)$, переписывающую содержимое текстового файла $t2$ в текстовый файл $t1$ (с сохранением деления на строки).

15.37. Описать процедуру $\text{присв}(t1, t2)$, переписывающую в текстовый файл $t1$ содержимое текстового файла $t2$, но без пустых строк.

15.38. Считая, что непустой текстовый файл f разбит на строки, длина каждой из которых не превосходит 80, описать процедуру $\text{преобр}(f, f80)$, которая, дополняя краткие строки файла f пробелами справа, формирует текстовый файл $f80$, все строки в котором имеют длину 80.

15.39*.

type слово = packed array [1..20] of char;
список = array [1..100] of слово;

Описать процедуру $зап(l, t)$, записывающую слова списка l как строки в текстовый файл t .

15.40*. В текстовом файле t записана непустая последовательность вещественных чисел, разделенных пробелами. Описать функцию $max(t)$ для нахождения наибольшего из этих чисел.

15.41. В текстовом файле $t1$ записана последовательность целых чисел, разделенных пробелами. Описать процедуру $positive(t1, t2)$, записывающую в текстовый файл $t2$ все положительные числа из $t1$.

15.42*. type data = record
число:1..31;
месяц:1..12;
год:1900..1999
end;

var d: data;

Напечатать дату d в следующем виде: 25.10.1917, 22.6.1941, 9.5.1945 и т. п.

15.43. Описать процедуру $lines(t)$, которая построчно печатает содержимое непустого текстового файла t , вставляя в начало каждой печатаемой строки ее порядковый номер (он должен занимать 4 позиции) и пробел.

15.44. Программа. Напечатать таблицу значений функций $\sin x$ и $\lg x$ на отрезке $[0,3]$ с шагом 0.1. Значения x печатать с одной цифрой в дробной части, значения синуса — с пятью, а значения тангенса — в экспоненциальной форме.

15.45. Программа. Напечатать первые 10 строк «треугольника Паскаля» в следующем виде:

a) 1
 1 1
 1 2 1
 1 3 3 1
1 4 6 4 1
...
1 9 ... 126 ... 9 1

б) 1
 1 1
 1 2 1
 1 3 3 1
1 4 6 4 1
...
1 9 ... 9 1

(В этом «треугольнике» крайние числа равны 1, а каждое внутреннее — сумме двух чисел над ним.)

15.46. Программа. Напечатать картинку, изображающую умножение «столбиком» двух заданных натуральных

чисел. Возможный пример:

$$\begin{array}{r} 39624 \\ \times \quad 8503 \\ \hline 118872 \\ + 198120 \\ 316992 \\ \hline 336922872 \end{array}$$

15.47*. Имеется внешний текстовый файл *BOOK*. Написать программу, которая, игнорируя исходное деление этого файла на строки, переформатирует его, разбивая на строки так, чтобы каждая строка оканчивалась точкой либо содержала ровно 60 литер, если среди них нет точки.

15.48. Программа. Имеется внешний текстовый файл *T*. Напечатать первую из самых коротких его строк.

15.49. Имеется внешний файл *KURS1* типа *курс*, содержащий сведения о студентах первого курса:

```
type строка = packed array [1..12] of char;
экзамен = (анализ,алгебра,программирование);
студент = record ФИО:record фам,имя,отч:
                                         строка end;
                                         оценки:array [экзамен] of 2..5;
                                         группа:101..116
                                         end;
курс = file of студент;
```

Написать программу, которая оставляет в файле *KURS1* сведения только о тех студентах, которые успешно сдали все экзамены, и выводит на печать сведения о студентах, имеющих хотя бы одну задолженность: печатает их фамилии и инициалы, номера их групп и количество несданных экзаменов.

15.50. Программа. Имеется внешний файл *A* из целых чисел. Используя еще три внешних файла *B*, *C* и *D* в качестве рабочих, упорядочить файл *A* по неубыванию следующим методом (называемым *внешней сортировкой сбалансированным слиянием*).

Назовем «отрезком» как можно более длинный упорядоченный по неубыванию участок файла (на рис. 7, а показан пример файла *A*, отрезки которого разделены вертикальными линиями). На начальном этапе сортировки определяются отрезки файла *A* и они попеременно переносятся в файлы *C* и *D* (рис. 7, б). На следующем этапе пары *i*-х отрезков файлов *C* и *D* (*i*=1, 2, ...) сливаются

в более длинные отрезки и попеременно переносятся в файлы *A* и *B* (рис. 7, *a*). Затем сливаются пары отрезков файлов *A* и *B* и переносятся в файлы *C* и *D* (рис. 7, *б*)

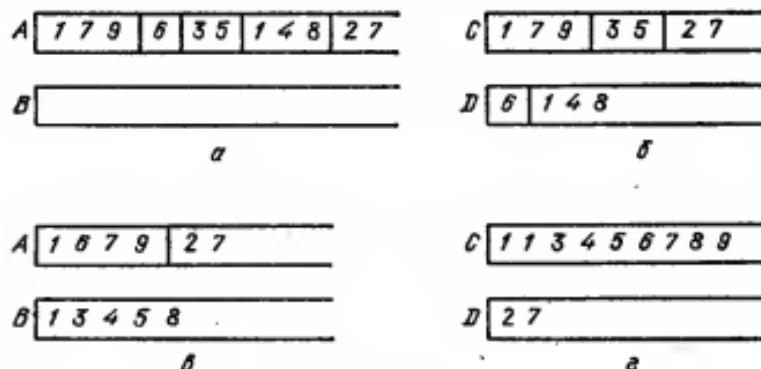


Рис. 7

и т. д. (Учесть, что в конце концов единая упорядоченная последовательность чисел должна оказаться в файле *A*.)

16. ССЫЛОЧНЫЕ ТИПЫ. СПИСКИ

16.1. type ref = ^integer;
var p,q:ref;

Пусть переменные *p* и *q* имеют значения, показанные на рис. 8. Ответить на следующие вопросы.

а) Что является значением переменной *p*: ссылка на объект (переменную) целого типа или сам этот объект?



Рис. 8

Что обозначает переменная *p*: ссылку на объект целого типа, сам этот объект или целое 5? Каковы типы переменных *p* и *p*?

б)* Что будет выдано на печать в результате выполнения следующих операторов?

```

p:=q;
if p=q then p:=nil else if p=q then q:=p;
if p=q then q:=4;
writeln(p);
  
```

16.2*. type D = record a:boolean; b,c:real end;
var r:D;

Пусть переменная *r* имеет значение, показанное на рис. 9.
Нарисовать структуру значения переменной *r* после выполнения следующих операторов:

If *r^.b*<>nil then *r^.c:=r^.b*;
r^.b:=r^.c-1.4; *r^.a:=r^.b=r^.c*

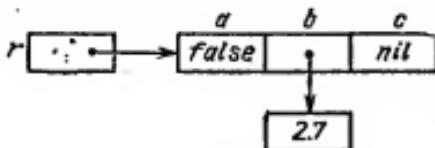


Рис. 9

16.3*. var p,q:tinteger; r:char;

Какие из следующих операторов неправильные и почему?

- а) *p:=q*; б) *q:=r*; в) *p:=nil*;
- г) *r:=nil*; д) *q:=p^*; е) *p^:=nil*;
- ж) *r^:=p^*; з) *q^:=ord(r^)*;
- и) if *r<>nil* then *r^:=nil^*;
- к) if *q>nil* then *q^:=p^*;
- л) if *q=p* then *write(q)*;
- м) if *q<>r* then *read(r^)*

16.4. Имеется программа

```
program dynamic (output);
  var x:boolean; y:boolean;
begin {A} new(x); {B} x^:=true; y:=not x^;
      {C} dispose(x); {D} writeln(y)
end.
```

Ответить на следующие вопросы.

- а) Какие переменные существуют в каждой из точек А, В, С и D и каковы их значения в эти моменты?
- б) Почему объекты (переменные), создаваемые процедурой *new* и уничтожаемые процедурой *dispose*, называют динамическими? Почему им не дают имена?
- в) Можно ли переменной *x* присвоить ссылку на переменную *y*? Можно ли с помощью процедуры *dispose* уничтожить переменные *x* и *y*?

16.5*. type A = tchar; B = record f1:tchar; f2:A end;
var p:B; q:A;

Нарисовать структуру значений переменных *p* и *q* после выполнения следующих операторов!

```
new(q); q^.:= '7';
new(p); p^.f1 := succ(q^.); p^.f2 := q
```

16.6. type chain = ^elem;
elem = record data:integer;
link:chain end;

```
var p,q:chain;
```

Нарисовать структуру значения переменной *p* после выполнения следующих операторов:

- а) new(p); p^.data:=4; p^.link:=nil;
- б) new(p); p^.data:=7; p^.link:=p;
- в) new(q); q^.data:=2; q^.link:=nil;
- new(p); p^.data:=1; p^.link:=q;

г)* new(p); p^.data:=5; new(p^.link); p^.link^.:=p†

16.7. Найти ошибки в следующей программе:

```
program errors (input,output);
var a,b:^integer;
begin if a=nil then read(a); a^.:=5;
b:=nil; b^.:=2;
new(b); read(b^.); writeln(b,b^.);
new(a); b^.:=a; dispose(a); b^.:=4
end.
```

16.8*. Почему недопустимы следующие описания и как их исправить?

```
type A = ^0..9;
B = record p:real; q:C end;
C = ^B;
```

16.9*. Описать переменную *p* (и, если надо, вспомогательные переменные) и выполнить операторы, присваивающие ей указанные значения (рис. 10).

16.10. type цепочка = ^звено;
звено = record элем:integer;
след:цепочка end;

```
var р:цепочка;
```

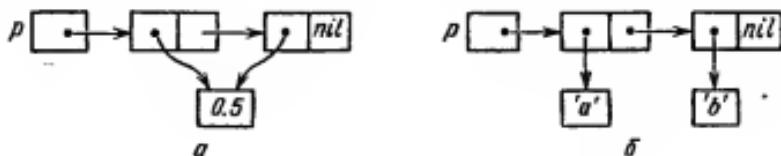
Выписать операторы, которые преобразуют значение переменной *p*, показанное на рис. 11,*a*, к значению, показанному на рис. 1)* 11,*b*; 2) 11,*c*; 3) 11,*g*. (Звенья, ставшие ненужными, уничтожить.)

16.11. Допустимы ли в языке Паскаль конструкции *p†[2]*, *q† + [2]* и *r††?* Ответ обосновать.

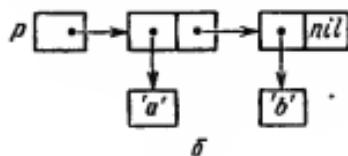
16.12. type ссылка = ^real;
вектор = array [1..100] of ссылка;

Считая, что все элементы вектора x отличны от nil, описать:

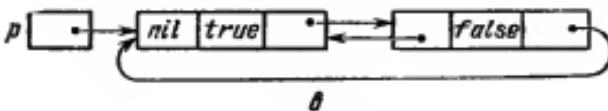
а) функцию $\max(x)$ для нахождения наибольшего из чисел, на которые ссылаются элементы вектора x ;



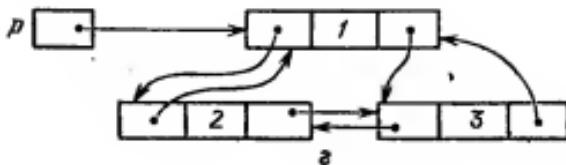
а



б



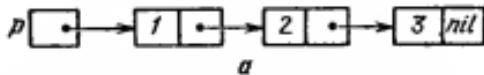
в



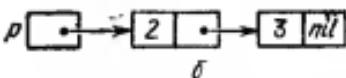
г

Рис. 10

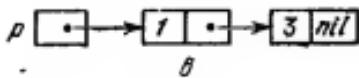
б)* функцию $\neg \text{el}(x)$, значением которой является первый из элементов вектора x , ссылающихся на отрицательные числа, или nil, если таких элементов нет;



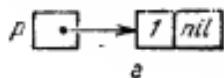
а



б



в



г

Рис. 11

в) логическую функцию $\text{same}(x)$, которая проверяет, есть ли в векторе x хотя бы две одинаковые ссылки;

г) процедуру $\text{unique}(x)$, которая в векторе x все элементы, ссылающиеся на равные числа, заменяет на первый из этих элементов.

16.13. Одно из возможных представлений «длинного» текста — это разделить его на участки (строки) равной длины и создать массив ссылок на эти строки:

```
const d=...; {длина строки}
n=...; {максим. число строк}
type строка = packed array [1..d] of char;
ссылка = ^строка;
текст = array [1..n] of ссылка;
```

(Если в тексте менее n строк, то последние элементы массива равны nil; в начале массива ссылок nil не должно быть. Если в операции над текстом указан номер отсутствующей строки, т. е. элемент массива с этим номером равен nil, то такая операция не выполняется.)

Используя данное представление текста, описать:

а) функцию *числострок*(T) для подсчета числа строк в тексте T ;

б) логическую функцию *элем*(T, i, j, c), проверяющую, есть ли в тексте T строка с номером i , и, если есть, присваивающую j -ю литеру этой строки параметру c ;

в) процедуру *перестановка*(T, i, j), меняющую местами i -ю и j -ю строки текста T ;

г) процедуру *замена*(T, i, j), заменяющую i -ю строку текста T на копию j -й строки;

д) процедуру *добавить*(T, i, j), добавляющую после i -й строки текста T копию j -й строки;

е) процедуру *удалить*(T, i), удаляющую i -ю строку из текста T ;

ж) логическую функцию *поиск*(T, c, i, j), определяющую, входит ли литера c в текст T , и, если входит, присваивающую параметрам i и j «координаты» первого вхождения этой литеры: i — номер строки, а j — номер позиции в этой строке;

з) процедуру *вывод*(T), печатающую построчно текст T ;

и) процедуру *ввод*(T),читывающую из входного файла последовательность литер до первой точки и формирующую из них текст T (последнюю строку, если надо, дополнить пробелами).

В упражнениях 16.14—16.29 использовать (линейные) односторонние списки без заглавного звена (рис. 12,а) или с заглавным звеном (рис. 12,б) при следующем их описании:

```
type ТЭ = ...; {тип элементов списка (уточняемый,
если надо, в упражнениях)}
```

список = †звено;
звено = record элем:ТЭ; след:список end;

При этом параметры L , L_1 и L_2 обозначают списки, а параметры E , E_1 и E_2 —данные типа ТЭ, к которым применимы операции присваивания и проверки на равенство.

16.14. Описать функцию или процедуру, которая:

- определяет, является ли список L пустым;
- находит среднее арифметическое элементов непустого списка L ($T\mathcal{E}=real$);
- * заменяет в списке L все вхождения E_1 на E_2 ;

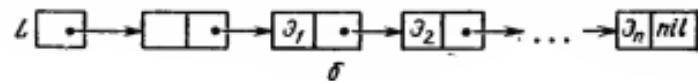
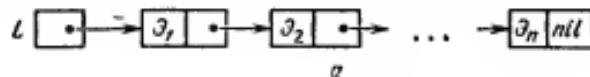


Рис. 12

г) меняет местами первый и последний элементы непустого списка L ;

д)* проверяет, упорядочены ли элементы списка L по алфавиту ($T\mathcal{E}='a'..'z'$);

е) находит сумму последнего и предпоследнего элементов списка L , содержащего не менее двух элементов ($T\mathcal{E}=integer$).

16.15. type слово=packed array [1..10] of char;
ТЭ=слово;

Описать функцию, подсчитывающую количество слов списка L , которые:

- начинаются и оканчиваются одной и той же литерой;
- начинаются с той же литеры, что и следующее слово;
- совпадают с последним словом.

16.16. type файл=file of ТЭ;
массив=аггр [1..50] of ТЭ;

Описать функцию, значением которой является список, построенный из элементов:

- * файла f ;
- массива x (список строить от конца).

16.17. Описать процедуру, которая по списку L строит два новых списка: L_1 —из положительных элементов и L_2 —из остальных элементов списка L ($T\mathcal{E}=real$).

16.18. Описать процедуру, которая вставляет

а) в начало списка L новый элемент E ;

б) в конец списка L новый элемент E ;

в) новый элемент E после первого элемента непустого списка L ;

г) в список L новый элемент $E1$ за каждым вхождением элемента E ;

д)* в список L новый элемент $E1$ перед первым вхождением элемента E , если E входит в L ;

е) в непустой список L пару новых элементов $E1$ и $E2$ перед его последним элементом;

ж) в непустой список L , элементы которого упорядочены по неубыванию, новый элемент E так, чтобы сохранилась упорядоченность ($T\Theta = real$).

16.19. Описать процедуру, которая удаляет

а) из непустого списка L первый элемент;

б) из списка L второй элемент, если такой есть;

в) из списка L за каждым вхождением элемента E один элемент, если такой есть и он отличен от E ;

г)* из непустого списка L последний элемент;

д) из списка L первый отрицательный элемент, если такой есть ($T\Theta = integer$);

е) из списка L все отрицательные элементы ($T\Theta = real$).

16.20*. Программа. Заданный во входном файле текст (за ним следует точка) распечатать в обратном порядке.

16.21. Программа. Даная непустая последовательность натуральных чисел, за которой следует 0. Напечатать порядковые номера тех чисел последовательности, которые имеют наибольшую величину.

16.22. Программа. Дано целое $n > 1$, за которым следует n вещественных чисел. Напечатать эти числа в порядке их неубывания.

16.23. Описать процедуру или функцию, которая:

а) проверяет на равенство списки $L1$ и $L2$;

б) определяет, входит ли список $L1$ в список $L2$;

в) проверяет, есть ли в списке L хотя бы два одинаковых элемента;

г) переносит в конец непустого списка L его первый элемент;

д) переносит в начало непустого списка L его последний элемент;

е)* добавляет в конец списка $L1$ все элементы списка $L2$;

ж) вставляет в список L за первым вхождением элемента E все элементы списка $L1$, если E входит в L ;

в) переворачивает список L , т. е. изменяет ссылки в этом списке так, чтобы его элементы оказались расположеными в обратном порядке;

и) в списке L из каждой группы подряд идущих равных элементов оставляет только один;

к) оставляет в списке L только первые вхождения одинаковых элементов.

16.24. Описать рекурсивную функцию или процедуру, которая:

а)* определяет, входит ли элемент E в список L ;

б) подсчитывает число вхождений элемента E в список L ;

в) находит максимальный элемент непустого списка L ($T\mathcal{E}=real$);

г) печатает в обратном порядке элементы списка L ($T\mathcal{E}=char$);

д) заменяет в списке L все вхождения $E1$ на $E2$;

е)* удаляет из списка L первое вхождение элемента E , если такое есть;

ж) удаляет из списка L все вхождения элемента E ;

з) строит $L1$ — копию списка L ;

и) удваивает каждое вхождение элемента E в список L ;

к) находит среднее арифметическое всех элементов непустого списка L ($T\mathcal{E}=real$).

16.25. Описать процедуру, которая формирует список L , включив в него по одному разу элементы, которые:

а) входят хотя бы в один из списков $L1$ и $L2$;

б) входят одновременно в оба списка $L1$ и $L2$;

в) входят в список $L1$, но не входят в список $L2$;

г) входят в один из списков $L1$ и $L2$, но в то же время не входят в другой из них.

16.26. Описать процедуру, которая объединяет два упорядоченных по неубыванию списка $L1$ и $L2$ ($T\mathcal{E}=real$) в один упорядоченный по неубыванию список:

а) построив новый список L ;

б) меняя соответствующим образом ссылки в $L1$ и $L2$ и присвоив полученный список параметру $L1$.

16.27. Описать процедуру *подстановка*($L, L1, L2$), которая в списке L заменяет первое вхождение списка $L1$ (если такое есть) на список $L2$.

16.28.

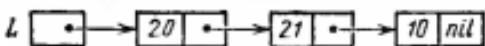
```
const n=...; {целая константа>1}
```

```
type число=packed array [1..n] of 0..9;
```

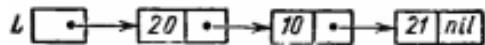
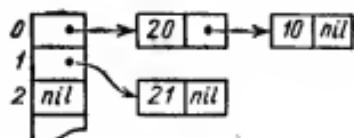
```
ТЭ=число;
```

Описать процедуру $упор(L)$, упорядочивающую по неубыванию числа непустого списка L с помощью следующего алгоритма (рис. 13, где p предполагается равным 2).

Создать 10 пустых подсписков (по количеству цифр), а затем, просматривая числа исходного списка, занести в k -й под список все числа, оканчивающиеся цифрой k , после чего эти под списки объединить в один список L ,



Шаг 1



Шаг 2

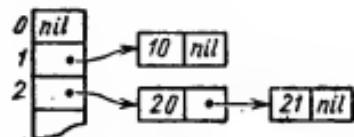


Рис. 13

записав в последнее звено k -го под списка ссылку на начало $(k+1)$ -го под списка. Далее аналогичный метод применяется по отношению к предпоследней цифре чисел (не нарушая при этом упорядоченность по последней цифре), затем—по отношению к третьей от конца цифре и т. д.

16.29. type слово=цепочка;

цепочка=record буква: 'а'..'з';

связь: слово end;

ТЭ=слово;

Описать функцию или процедуру, которая:

а) в списке L переставляет местами первое и последнее непустые слова, если в L есть хотя бы два непустых слова;

б)* печатает текст из первых букв всех непустых слов списка L ;

- в) удаляет из непустых слов списка L их первые буквы;
 г) печатает все непустые слова списка L ;
 д) определяет количество слов в непустом списке L ,
 отличных от последнего.

16.30. Многочлен

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

с целыми коэффициентами можно представить в виде списка (рис. 14, а), причем если $a_i = 0$, то соответствующее звено

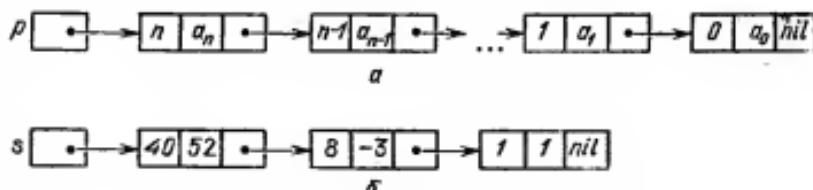


Рис. 14

не включается в список (на рис. 14, б показано представление многочлена $S(x) = 52x^4 - 3x^3 + x$).

Описать на Паскале тип данных, соответствующий такому представлению многочленов, и определить следующие функции и процедуры для работы с этими списками-многочленами:

- логическую функцию *равно*(p, q), проверяющую на равенство многочлены p и q ;
- функцию *знач*(p, x), вычисляющую значение многочлена p в целочисленной точке x ;
- процедуру *диф*(p, q), которая строит многочлен p — производную многочлена q ;
- процедуру *слож*(p, q, r), которая строит многочлен p — сумму многочленов q и r ;
- процедуру *вывод*(p, v), которая печатает многочлен p как многочлен от переменной, однобуквенное имя которой является значением логического параметра v ; например, для указанного выше многочлена S процедура *вывод*($s, 'y'$) должна напечатать

$$52y^4 - 3y^3 + y;$$

- процедуру *ввод*(p), которая считывает из входного файла безошибочную запись многочлена (за ней — пробел) и формирует соответствующий список-многочлен p .

16.31. Предложить и описать на Паскале представление файлов (из элементов некоторого типа *ТЭ*) в виде списков и определить функцию *eof1(f)* и процедуры *reset1(f)*,

readl(f,x), *rewritel(f)* и *writel(f,x)*, реализующие при таком представлении файлов действия соответствующих стандартных функций и процедур.

16.32. Назовем (иерархическим) «списком» заключенную в круглые скобки последовательность элементов, разделенных запятыми; элементы списка—это атомы или снова списки:

```
<список> ::= ( ) | <элементы>
<элементы> ::= <элемент> | <элемент>, <элементы>
<элемент> ::= <атом> | <список>
```

Под «атомом» понимается последовательность, содержащая от 1 до *n* букв и цифр, где *n*—заранее известное натуральное число. Пример подобного списка: (AD75,(3,(),(7H))).

Предложить и описать на Паскале представление таких списков и определить следующие рекурсивные функции и процедуры для работы с ними:

- логическую функцию *member(A,L)*, проверяющую, входит ли атом *A* в список *L*;
- логическую функцию *equal(L1,L2)*, проверяющую на равенство списки *L1* и *L2*;
- процедуру *printat(L)*, печатающую все атомы, входящие в список *L*;
- процедуру *printlist(L)*, печатающую список *L* в том виде, как он определен указанными выше металингвистическими формулами;
- процедуру *readlist(L)*, которая считывает из входного файла записанный без ошибок список и строит *L*—соответствующее представление этого списка.

16.33. Пусть *L* обозначает кольцевой (циклический) двунаправленный список с заглавным звеном (рис. 15) при следующем описании такого списка:

```
тире ТЭ2=...; {тип элементов списка}
список2=звено2;
звено2=record элем:ТЭ2;
пред,след:список2 end;
```

и пусть *E* обозначает величину типа ТЭ2.

Описать функцию или процедуру, которая:

- * определяет, является ли список *L* пустым;
- * печатает в обратном порядке элементы непустого списка *L* (*ТЭ2=char*);
- подсчитывает количество элементов списка *L*, у которых равные «соседи»;
- определяет, есть ли в списке *L* хотя бы один эле-

мент, который равен следующему за ним (по кругу) элементу;

д) в списке L переставляет в обратном порядке все элементы между первым и последним вхождениями элемента E , если E входит в L не менее двух раз;

е) удаляет из списка L первый отрицательный элемент, если такой есть;

ж) из списка L , содержащего не менее двух элементов, удаляет все элементы, у которых одинаковые «соседи» (первый и последний элементы считать соседями);

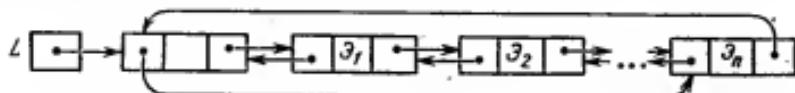


Рис. 15

з) добавляет в конец списка L новый элемент E ;

и) в списке L удваивает каждое вхождение элемента E ;

к) строит список L по одностороннему списку L_1 ;

л) в конец непустого списка L добавляет все его элементы, располагая их в обратном порядке (например, по списку из элементов 1, 2, 3 требуется построить список из элементов 1, 2, 3, 3, 2, 1);

16.34. («Считалка».) n ребят располагаются по кругу. Начав отсчет от первого, удаляют каждого $k-го, смыкая круг после каждого удаления. Определить порядок удаления ребят из круга.$

Решение этой задачи описать в виде программы (ее исходные данные—натуральные числа n и k), которая должна напечатать номера ребят в том порядке, как они удаляются из круга.

Решения задач 16.35—16.46 описать в виде программ, выбрав для представления данных подходящую списковую структуру.

16.35. Определить, симметричен ли заданный во входном файле текст (за ним следует точка).

16.36. Даная последовательность из не менее чем двух различных натуральных чисел, за которой следует 0. Напечатать в обратном порядке все числа между наибольшим и наименьшим числами этой последовательности.

16.37. Даная непустая последовательность непустых слов из букв; между соседними словами—запятая, за последним словом—точка. Напечатать все слова максимальной длины.

16.38. Даны непустая последовательность слов, в каждом из которых от 1 до 12 строчных латинских букв; между словами — пробел, за последним словом — точка. Напечатать эти слова по алфавиту, указав для каждого из них число его вхождений в эту последовательность.

16.39. Решить предыдущую задачу в предположении, что максимальная длина слов заранее не известна.

16.40. Даны непустая последовательность слов, в каждом из которых от 1 до 8 строчных латинских букв; между словами — пробел, за последним словом — точка. Напечатать эти слова в следующем порядке: сначала — по алфавиту все слова из одной буквы, затем — по алфавиту все двухбуквенные слова и т. д. (одинаковые слова печатать по одному разу).

16.41. Решить предыдущую задачу в предположении, что максимальная длина слов заранее не известна.

16.42. Даи текст, оканчивающийся точкой. Среди литер этого текста особую роль играет знак $\#$, появление которого в тексте означает отмену предыдущей литеры текста; k знаков $\#$ подряд отменяют k предыдущих литер (если такие есть). Напечатать данный текст, исправленный с учетом такой роли знака $\#$ (например, текст ХЭ#Е##HELO#LO должен быть напечатан в виде HELLO).

16.43. Дано произвольное натуральное число n . Напечатать все цифры десятичной записи числа $n!$

16.44. Дано целое $n > 2$. Напечатать коэффициенты n -го многочлена Чебышева $T_n(x)$, определяемого формулами

$$T_0(x) = 1; \quad T_1(x) = x; \quad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \quad (k=2, 3, \dots).$$

16.45. Даны запись многочлена (от переменной x) произвольной степени с целыми коэффициентами, причем его одночлены могут быть и не упорядочены по степеням x , а одночлены одной и той же степени могут повторяться. Возможный пример

$$-8x^{14} - 74x^{14} + 8x^{14} + 5 - x^{13}.$$

Требуется привести подобные члены в этом многочлене, после чего распечатать его по убыванию степеней x .

16.46. Промоделировать выполнение любого заданного нормального алгоритма Маркова над любым заданным входным словом (см., например, [9]).

17. ОЧЕРЕДИ, СТЕКИ, ДВОИЧНЫЕ ДЕРЕВЬЯ

17.1. Для работы с очередью, т. е. последовательностью элементов, в которую элементы всегда добавляются в конец, а удаляются из начала («первым пришел—первым ушел»), нужны обычно следующие операции:

ОЧИСТОЧ(Q)—создать пустую очередь *Q* (очистить очередь);

ПУСТОЧ(Q)—проверить, является ли очередь *Q* пустой;

ВОЧЕРЕДЬ(Q,x)—добавить в конец очереди *Q* элемент *x*;

ИЗОЧЕРЕДИ(Q,x)—удалить из очереди *Q* первый элемент, присвоив его параметру *x*.

Требуется для каждого из указанных ниже представлений очереди описать на Паскале соответствующий тип *очередь*, считая, что все элементы очереди имеют некоторый тип *ТЭО*, и реализовать в виде процедур или функций перечисленные операции над очередью (если операция по тем или иным причинам не может быть выполнена, следует передать управление некоторой процедуре *ОШИБКА(k)*, считая ее уже описанной, где *k*—номер ошибки: 1—переполнение очереди, 2—исчерпание очереди).

Представление очереди (*n*—целая константа > 1):

а)* для каждой очереди отводится свой массив из *n* компонент типа *ТЭО*, в котором элементы очереди занимают группу соседних компонент, индексы первой и последней из которых запоминаются (рис. 16, *a*); при этом, когда очередь достигает правого края массива, все ее элементы сдвигаются к левому краю;

б) аналогичное представление, но массив как бы склеивается в кольцо, поэтому если очередь достигает правого края массива, то новые элементы записываются в начало массива (рис. 16, *b*);

в) для каждой очереди создается свой односторонний список из элементов типа *ТЭО*, при этом запоминаются ссылки на первое и последнее звенья списка (рис. 16, *c*).

17.2. Используя очередь (считать уже описанными тип *очередь* при подходящем типе *ТЭО*, функцию *ПУСТОЧ* и процедуры *ОЧИСТОЧ*, *ВОЧЕРЕДЬ* и *ИЗОЧЕРЕДИ*—см. 17.1), решить следующую задачу (решение описать в виде процедуры).

а)* type FR=file of real;

За один просмотр файла *f* типа *FR* и без использования дополнительных файлов напечатать элементы файла *f*

в следующем порядке: сначала — все числа, меньшие a , затем — все числа из отрезка $[a, b]$, и наконец — все остальные числа, сохраняя исходный взаимный порядок в каждой из этих трех групп чисел (a и b — заданные числа, $a < b$).

б) Содержимое текстового файла f , разделенное на строки, переписать в текстовый файл g , перенося при этом

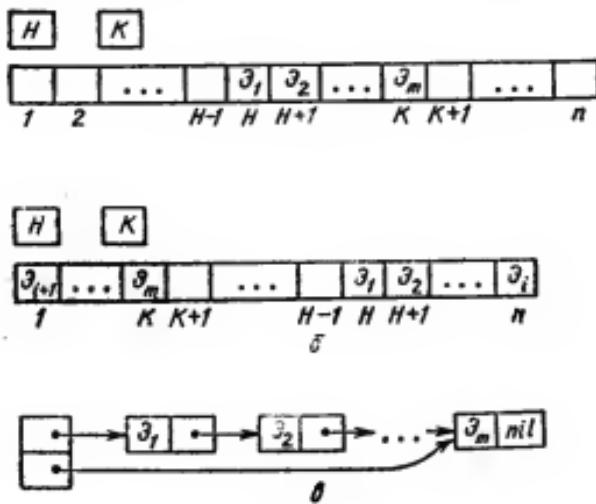


Рис. 16

в конец каждой строки все входящие в нее цифры (с сохранением исходного взаимного порядка как среди цифр, так и среди остальных литер строки).

в) type имя = (Анна, ..., Яков);
дети = packed array [имя, имя]
of boolean;
потомки = file of имя;

Считая заданными имя I и массив D типа $дети$ ($D[x,y]=true$, если человек по имени y является ребенком человека по имени x), записать в файл P типа $потомки$ имена всех потомков человека с именем I в следующем порядке: сначала — имена всех его детей, затем — всех его внуков, затем — всех правнуков и т. д.

17.3. Для работы со стеком, т. е. последовательностью элементов, в которой элементы всегда добавляются в конец и удаляются из конца («последним пришел — первым ушел»), нужны обычно следующие операции:

ОЧИСТЕК(S) — создать пустой стек S (очистить стек);
ПУСТЕК(S) — проверить, является ли стек S пустым;

BСTEK(S,x)—добавить в конец стека *S* элемент *x*;
ИЗСТЕКА(S,x)—удалить из очереди *S* последний элемент, присвоив его параметру *x*.

Требуется для каждого из указанных ниже представлений стека описать на Паскале соответствующий тип *стек*, считая, что все элементы стека имеют некоторый тип *TЭC*, и реализовать в виде процедур или функций перечисленные операции над стеком (если операция по тем или иным причинам невыполнима, следует передать управление некоторой процедуре *ОШИБКА(k)*, считая ее уже описанной, где *k*—номер ошибки: 1—переполнение стека 2—исчерпание стека).

Представление стека (*n*—целая константа > 1):

а) для каждого стека отводится свой массив из *n* компонент типа *TЭC*, в начале которого располагаются элементы стека, при этом запоминается индекс компонента массива, занятой последним элементом стека (рис. 17, а):

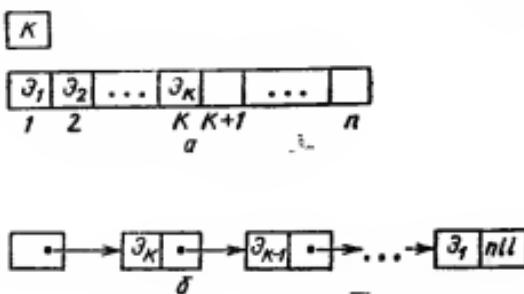


Рис. 17

б) для каждого стека создается свой односторонний список, в котором элементы стека располагаются в обратном порядке (рис. 17, б).

17.4. Используя стек (считать уже описанными тип *стек* при *TЭC=char*, функцию *ПУСТЕК* и процедуры *ОЧИСТЕК*, *ВСТЕК* и *ИЗСТЕКА*—см. 17.3), решить следующую задачу (решение описать в виде процедуры или функции).

а) Напечатать содержимое текстового файла *t*, выписывая литеры каждой его строки в обратном порядке.

б) Проверить, является ли содержимое текстового файла *t* правильной записью формулы следующего вида:

Формула ::= *терм* | *терм* + *формула* |
 терм — *формула*

$\langle\text{терм}\rangle ::= \langle\text{имя}\rangle \mid (\langle\text{формула}\rangle) \mid [\langle\text{формула}\rangle] \mid$
 $\quad \langle\text{формула}\rangle$
 $\langle\text{имя}\rangle ::= x \mid y \mid z$

в)* В текстовом файле f записана без ошибок формула следующего вида:

$\langle\text{формула}\rangle ::= \langle\text{цифра}\rangle \mid M(\langle\text{формула}\rangle, \langle\text{формула}\rangle) \mid$
 $\quad m(\langle\text{формула}\rangle, \langle\text{формула}\rangle)$
 $\langle\text{цифра}\rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

где M обозначает функцию \max , а m — \min .

Вычислить (как целое число) значение данной формулы (например, $M(5, m(6, 8)) \rightarrow 6$).

г) В текстовом файле LOG записано без ошибок логическое выражение ($ЛВ$) в следующей форме:

$\langle\text{ЛВ}\rangle ::= \text{true} \mid \text{false} \mid (\neg \langle\text{ЛВ}\rangle) \mid (\langle\text{ЛВ}\rangle \wedge \langle\text{ЛВ}\rangle) \mid$
 $\quad (\langle\text{ЛВ}\rangle \vee \langle\text{ЛВ}\rangle)$

где знаки \neg , \wedge и \vee обозначают соответственно отрицание, конъюнкцию и дизъюнкцию.

Вычислить (как *boolean*) значение этого выражения.

17.5. Используя очередь и/или стек (считать уже описанными их типы и операции над ними — см. 17.1 и 17.3), решить следующую задачу (решение описать в виде процедуры).

В текстовом файле f записан текст, сбалансированный по круглым скобкам:

$\langle\text{текст}\rangle ::= \langle\text{пусто}\rangle \mid \langle\text{элемент}\rangle \langle\text{текст}\rangle$
 $\langle\text{элемент}\rangle ::= \langle\text{буква}\rangle \mid (\langle\text{текст}\rangle)$

Требуется для каждой пары соответствующих открывающей и закрывающей скобок напечатать номера их позиций в тексте, упорядочив пары номеров в порядке возрастания номеров позиций:

- закрывающих скобок;
- открывающих скобок.

Например, для текста $A + (45 - F(X)*(B - C))$ надо напечатать:

- 8 10; 12 16; 3 17;
- 3 17; 8 10; 12 16.

17.6. Под «выражением» будем понимать конструкцию следующего вида:

$\langle\text{выражение}\rangle ::= \langle\text{терм}\rangle$
 $\quad \langle\text{терм}\rangle \langle\text{знак}+\rangle \rightarrow \langle\text{выражение}\rangle$
 $\langle\text{знак}+\rangle ::= + \mid -$

<терм> ::= <множитель> | <множитель> * <терм>
 <множитель> ::= <число> | <переменная> | (<выражение>) |
 <множитель> \dagger <число>
 <число> ::= <шифра>
 <переменная> ::= <буква>

где знак \dagger обозначает операцию возведения в степень.

Постфиксной формой записи выражения $a \Delta b$ называется запись, в которой знак операции размещён за операндами $ab \Delta$. Примеры:

$$\begin{array}{ll}
 a - b & \rightarrow ab - \\
 a * b + c & \rightarrow ab * c + \quad (\text{т. е. } (ab *)c +) \\
 a * (b + c) & \rightarrow abc + * \quad (\text{т. е. } a(bc +)*) \\
 a + b \dagger c \dagger d * e & \rightarrow abc \dagger d \dagger e * +
 \end{array}$$

а) Описать функцию *value(postfix)*, которая вычисляет как целое число значение выражения (без переменных), записанного в постфиксной форме в текстовом файле *postfix*.

Использовать следующий алгоритм вычисления. Выражение просматривается слева направо. Если встречается операнд (число), то его значение (как целое) заносится в стек, а если встречается знак операции, то из стека извлекаются два последних элемента (это операнды данной операции), над ними выполняется операция и ее результат записывается в стек. В конце концов в стеке остается только одно число — значение всего выражения.

б) Описать процедуру *translate(infix, postfix)*, которая переводит выражение, записанное в обычной (инфиксной) форме в текстовом файле *infix*, в постфиксную форму и в таком виде записывает его в текстовый файл *postfix*.

Использовать следующий алгоритм перевода. В стек записывается открывающая скобка, и выражение просматривается слева направо. Если встречается операнд (число или переменная), то он сразу переносится в файл *postfix*. Если встречается открывающая скобка, то она заносится в стек, а если встречается закрывающая скобка, то из стека извлекаются находящиеся там знаки операций до ближайшей открывающей скобки, которая также удаляется из стека, и все эти знаки (в порядке их извлечения) записываются в файл *postfix*. Когда же встречается знак операции, то из конца стека извлекаются (до ближайшей скобки, которая сохраняется в стеке) знаки операций, старшинство которых больше или равно старшинству данной операции, и они записываются в файл *postfix*, после чего рассматриваемый знак заносится в стек. В за-

ключение выполняются такие же действия, как если бы встретилась закрывающая скобка.

в) Описать (нерекурсивную) процедуру *infixprint(postfix)*, которая печатает в обычной (инфиксной) форме выражение, записанное в постфиксной форме в текстовом файле *postfix*. (Лишие скобки желательно не печатать.)

В упражнениях 17.7—17.14 использовать двоичные деревья (рис. 18) при следующем их описании

```
type ТЭД=...; {тип элементов дерева}
дерево=†вершина;
вершина=record элем:ТЭД;
лев,прав:дерево end;
```

В этих упражнениях *T*, *T1* и *T2* обозначают деревья, а *E*—величину типа ТЭД.

17.7. Используя очередь или стек (считать уже описанными их типы и операции над ними—см. 17.1 и 17.3), описать процедуру или функцию, которая:

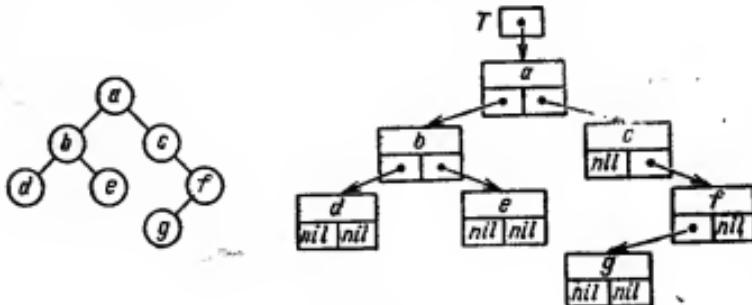


Рис. 18

а) присваивает параметру *E* элемент из самого левого листа непустого дерева *T* (лист—вершина, из которой не выходит ни одной ветви);

б)* определяет число вхождений элемента *E* в дерево *T*;

в) вычисляет среднее арифметическое всех элементов непустого дерева *T* (*ТЭД=real*);

г) заменяет в дереве *T* все отрицательные элементы на их абсолютные величины (*ТЭД=real*);

д) меняет местами максимальный и минимальный элементы непустого дерева *T*, все элементы которого различны (*ТЭД=real*);

е) печатает элементы из всех листьев дерева *T* (*ТЭД=char*);

ж)* печатает все элементы дерева T по уровням сначала—из корня дерева, затем (слева направо)—из вершин, дочерних по отношению к корню, затем (также слева направо)—из вершин, дочерних по отношению к этим вершинам, и т. д. ($T\text{ЭД}=integer$);

з) находит в непустом дереве T длину (число ветвей) пути от корня до ближайшей вершины с элементом E ; если E не входит в T , за ответ принять -1 ;

и) подсчитывает число вершин на l -ом уровне непустого дерева T (корень считать вершиной 0-го уровня).

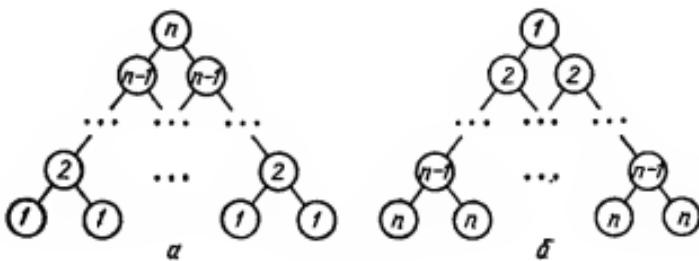


Рис. 19

17.8. Описать рекурсивную функцию или процедуру, которая:

а) определяет, входит ли элемент E в дерево T ;

б)* определяет число вхождений элемента E в дерево T ;

в) вычисляет сумму элементов непустого дерева T ($T\text{ЭД}=real$);

г) находит величину наибольшего элемента непустого дерева T ($T\text{ЭД}=real$);

д) печатает элементы из всех листьев дерева T ($T\text{ЭД}=char$);

е)* определяет максимальную глубину непустого дерева T , т. е. число ветвей в самом длинном из путей от корня дерева до листьев;

ж) подсчитывает число вершин на l -ом уровне непустого дерева T (корень считать вершиной 0-го уровня).

17.9. Рекурсивно и нерекурсивно описать логическую функцию $equal(T_1, T_2)$, проверяющую на равенство деревья T_1 и T_2 .

17.10*. Описать процедуру $copy(T, T_1)$, которая строит T_1 — копию дерева T .

17.11. Описать процедуру $create(T, l)$, где l — положительное целое число, которая строит T — дерево, показанное на рис. 19.

17.12. Описать логическую функцию *same(T)*, определяющую, есть ли в дереве *T* хотя бы два одинаковых элемента.

17.13. Формулу вида

«формула»:=<терминал> |
<формула> <знак> <формула>
<знак>:=+ | - | *
<терминал>:=0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

можно представить в виде двоичного дерева («дерева-формулы») с *ТЭД=char* согласно следующим правилам: формула из одного терминала (цифры) представляется деревом из одной вершины с этим терминалом, а формула вида

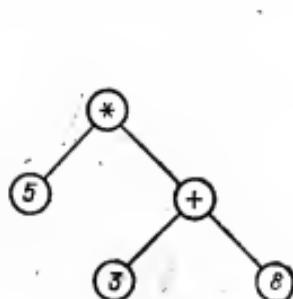


Рис. 20

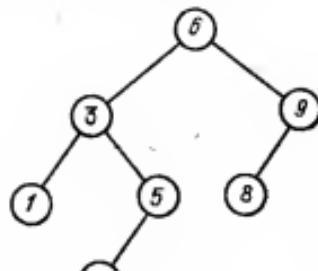


Рис. 21

$(f_1 \# f_2)$ — деревом, в котором корень — это знак *#*, а левое и правое поддеревья — это соответствующие представления формул f_1 и f_2 . (На рис. 20 показано дерево-формула, соответствующее формуле $(5*(3+8))$.)

Описать рекурсивную функцию или процедуру, которая:

- вычисляет (как целое число) значение дерева-формулы *T*;

- по формуле из текстового файла *f* строит соответствующее дерево-формулу *T*;

- печатает дерево-формулу *T* в виде соответствующей формулы;

- роверяет, является ли двоичное дерево *T* деревом-формулой.

17.14. Пусть в дереве-формулё (см. упр. 17.13) в качестве терминалов используются не только цифры, но и буквы, играющие роль переменных. Описать процедуру, которая:

- упрощает дерево-формулу *T*, заменяя в нем все поддеревья, соответствующие формулам $(f+0)$, $(0+f)$, $(f-0)$,

$(f * 1)$ и $(1 * f)$, на поддеревья, соответствующие формуле f , а поддеревья, соответствующие формулам $(f * 0)$ и $(0 * f)$, — на вершину с 0;

б) преобразует дерево-формулу T , заменяя в нем все поддеревья, соответствующие формулам $((f_1 \pm f_2) * f_3)$ и $(f_1 * (f_2 \pm f_3))$, на поддеревья, соответствующие формулам $((f_1 * f_3) \pm (f_2 * f_3))$ и $((f_1 * f_2) \pm (f_1 * f_3))$;

в) выполняет в дереве-формуле T преобразования, обратные преобразованиям из п. б;

г) строит дерево-формулу $T1$ — производную дерева-формулы T по переменной, однобуквенное имя которой является значением литерного параметра v .

17.15. Предложить и описать на Паскале представление в виде двоичного дерева для формул из упр. 17.13, в которых в качестве терминалов используются любые неотрицательные целые числа. Для такого представления реализовать те же операции, что и в упр. 17.13.

17.16. Деревом поиска, или таблицей в виде дерева, называется двоичное дерево, в котором слева от любой вершины находятся вершины с элементами, меньшими элемента из этой вершины, а справа — с большими элементами (предполагается, что все элементы дерева попарно различны и что их тип (ТЭД) допускает применение операций сравнения); пример такого дерева показан на рис. 21.

Считая описанными тип *дерево* (см. выше) и тип *файл*:
type файл = file of ТЭД;

определить функцию или процедуру, которая:

а) проверяет, входит ли элемент E в дерево поиска T ;
б) записывает в файл f элементы дерева поиска T в порядке их возрастания;

в) добавляет к дереву поиска T новый элемент E , если его не было в T ;

г) по файлу f , все элементы которого различны, строит соответствующее дерево поиска T .

17.17. Во внешнем текстовом файле *PROG* записана (без ошибок) некоторая программа на языке Паскаль. Известно, что в этой программе каждый идентификатор (служебное слово или имя) содержит не более 9 латинских букв и/или цифр. Напечатать в алфавитном порядке все различные идентификаторы этой программы, указав для каждого из них число его вхождений в текст программы. (Учесть, что в идентификаторах одноименные прописные и строчные буквы отождествляются, что внутри литерных

значений, строк-коистант и комментариев последовательности из букв и цифр не являются идентификаторами и что в записи вещественных чисел может встретиться буква *E* или *e*.)

Для хранения идентификаторов использовать дерево поиска, элементами которого являются пары — идентификатор и число его вхождений в текст программы.

17.18. Решить предыдущую задачу, но вместе с каждым идентификатором печатать в возрастающем порядке номера всех строк текста программы, в которых он встречается.

17.19. Решить задачу 17.17 при условии, что максимальная длина идентификаторов заранее не известна.

17.20. Решить задачу 17.18 при условии, что максимальная длина идентификаторов заранее не известна.

ОТВЕТЫ И РЕШЕНИЯ

1. ЧИСЛОВЫЕ ТИПЫ. ОПЕРАТОР ПРИСВАИВАНИЯ

- 1.1. а) 120; б) 64; в) 6.38; г) -0.7444; д) 2.75; е) -0.1667;
ж) 1.4142; в) 3.1416; и) 5E6; к) -24.8E-7; л) 1E6; м) 1E-5.
- 1.2. а) -2.7; б) 0.666; в) 10.0.
- 1.3. Неправильные: в), г), е), ж), а), л), м).
- 1.6. Нет.
- 1.8. а) $a+b*x+c*y*z$; б) $((a*x-b)*x+c)*x-d$;
в) $a*b/c+c/(a*b)$; г) $(x+y)/a1*a2/(x-y)$;
- д) $1E4*alpha-3.2*beta$; е) $(1+x/2+y/6)/(1+2/(3+x*y))$
- 1.9. а) $\frac{p+q}{r+s} - \frac{pq}{rs}$; б) $10^3 + \frac{\beta}{x_2 - \gamma\delta}$.
- 1.10. 32.0
- 1.12. 7 операций; $(x+0.5)*(y+0.7)-0.75$
- 1.14. б) $\text{sqr}(1+\text{sqr}(x))$; д) $\text{sqr}(\cos(x*x*x))$;
- ж) $\ln(x/5)/\ln(2)$; к) $\arctan(x/\text{sqr}(1-\text{sqr}(x)))$
- 1.15. а) $1/x$; б) $\text{sqr}(\text{sqr}(x))$; д) $\exp(100*\ln(x))$;
- а) $\exp(\ln(1+x)/3)$
- 1.16. $\exp(1); 4*\arctan(1)$
- 1.17. $\sin(3.1415927*x/180)$
- 1.20. а) $y:=1+x*(1+x/2*(1+x/3*(1+x/4)))$
- 1.21. б) $d:=\text{sqr}(\text{sqr}(x1-x2)+\text{sqr}(y1-y2))$;
- г) $p:=(a+b+c)/2$; $d:=\text{sqr}(p*(p-a)*(p-b)*(p-c))$
- 1.25. г:=-x; x:=-y; y:=г
- 1.27. а) 6; б) 7; в) 6; г) 6; д) -1; е) -2; ж) 1; с) -1
- 1.28. $d:=x-\text{trunc}(x)$
- 1.29. а) 3; б) 2; в) 5; г) 0; д) 0; е) 2; ж) и з) — ошибки.
- 1.31. $\{-(a \bmod b)\} + ((a \bmod b) * c)$
- 1.32. б) 2.
- 1.33. Целый тип: в), ж), з).
- 1.35. Правильные: а), г), д).
- 1.36. а) Неправильный оператор: если k — вещественная переменная, тогда недопустима операция $k \bmod 3$, а если k — целая

переменная, то ей присваивается значение вещественного выражения ($\cos(0)$ —вещественное число), что также недопустимо.

1.37. $h:=k \text{ div } 100 \text{ mod } 10$

2. ЛОГИЧЕСКИЙ ТИП

2.1. а) true; б) false; в) true.

2.2. а) $k \bmod 7=0$; б) $\text{sqr}(b)-4*a*c < 0$;

в) $\text{sqr}(x-1)+\text{sqr}(y) > \text{sqr}(r)$; г) $\text{sqr}(\text{round}(\text{sqr}(n))) = n$

2.3. а), б), г)—true; в), д)—false.

2.4. а) $(x > 0) \text{ and } (x < 1)$; б) $(x >= y) \text{ and } (x >= z)$;

в) $(x < y) \text{ or } (x < z)$; г) а or b; д) а and b

2.5. а)

a	$\text{not } a$	$a \text{ and } (\text{not } a)$
true	false	false
false	true	false

2.6. Ошибка в каждом случае.

2.7. а), б)—недопустимые типы операндов; в)—операнды разных типов; г), д)—отношения не заключены в скобки (согласно

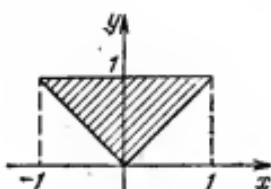


Рис. 22

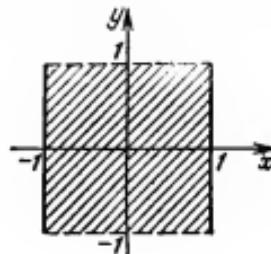


Рис. 23

синтаксису Паскаля выражение $\text{not } 2=5$ воспринимается как $(\text{not } 2)=5$, а выражение $x > 0 \text{ or } y=4$ —как $x > (0 \text{ or } y)=4$; е)—два знака or подряд.

2.8. а) $(a \text{ and } b) \text{ or } ((\text{not } c) \text{ and } b)$

2.10. в) $(x >= 2) \text{ and } (x <= 5) \text{ or } \text{abs}(x) <= 1$;

г) $(x < -1) \text{ or } (x > 1) \text{ and } (x < 2) \text{ or } (x > 5)$;

к) $(y \bmod 400=0) \text{ or } (y \bmod 4=0) \text{ and } (y \bmod 100 < > 0)$

2.11. а) см. рис. 22.

2.12. в) $(\text{sqr}(x)+\text{sqr}(y) <= 1) \text{ and } (\text{abs}(x)+\text{abs}(y) >= 1)$

2.13. а), г)—true; б), в)—false.

2.14. а) false; б) true; в) false.

2.15. а) $\text{odd}(n)=\text{odd}(k)$; б) $a < > b$ (или $\text{ord}(a)+\text{ord}(b)=1$).

2.16. а) см. рис. 23.

2.17. а)

a	b	$a \leq b$	$\text{not } a$	$\text{not } a \text{ or } b$
true	true	true	false	true
true	false	false	false	false
false	true	true	true	true
false	false	true	true	true

2.18. а) $\text{not } a \text{ and } b$

2.19. а) true.

3. ПРОСТЕЙШИЕ ПРОГРАММЫ

3.1. $x1=1.0 \quad x2=-2.0$

```
3.2. program коэффи (input, output);
      var x1,x2,p,q:real;
begin read(x1,x2); {авод корней}
      p:=-(x1+x2); q:=x1*x2; {вычисление
                                коэффициентов}
      writeln('коэфициенты:',1.0,p,q)
end.
```

3.5. Нельзя, так как исходным данным для целой переменной (a) не может быть вещественное число (5.0).

3.7. 3 2 3

3.8. $x+1$

3.10. program печать (input, output);

```
var a,a3,a6:integer;
begin read(a); a3:=a*a*a; a6:=sqr(a3);
      writeln(a); writeln(a3,a6); writeln(a6,a3,a)
end.
```

3.11. Достаточно в разделе констант число 2.71828 заменить на число 3.14159.

```
3.12. program больше (output);
      const e=2.71828; pi=3.14159;
begin writeln(exp(pi)>exp(e+ln(pi))) end.
```

3.15. а) В операторе присваивания $d:=\text{sqr}(d)$ слова указание вмв константы, что запрещено.

б) Применять функцию ord к вещественному аргументу (4) нельзя. В выражении $\text{ord}(x)=k$ сравниваются величины разных типов, что недопустимо.

в) Переменная c не получила значения, поэтому прибавлять к ней ничего нельзя.

г) Переменная y не описана.

д) Имя *B* описано дважды — как константа и как переменная (в именах одноименные заглавные и строчные буквы, в частности *B* и *b*, не различаются).

4. ОПЕРАТОРЫ: УСЛОВНЫЙ, СОСТАВНОЙ, ПУСТОЙ, ПЕРЕХОДА

- 4.1. а) if (*x*>0)and(*x*<2) then *y*:=sqr(cos(*x*))
else *y*:=1-sin(sqr(*x*));
б) if abs(*a*)<=3,14159/2 then *x*:=exp(sin(*a*)-1);
г) if (*a*>*b*)and(*a*>*c*) then *d*:=*a* else
if *b*>*c* then *d*:=*b* else *d*:=*c*;
д) if *x*<0 then if *x*>*y* then *z*:=*x* else *z*:=*y*
else if *x*<*y* then *z*:=*x* else *z*:=*y*
4.3. а) *c*:=*y* div 100; if *y* mod 100 <> 0 then *c*:=*c*+1;
б) *d*:=sqr(*b*)-4*a*c; *t*:=*d*>=0;
if *t* then begin *d*:=sqrt(*d*); *a2*:=2*a;
x1:=(-*b*+*d*)/*a2*; *x2*:=(-*b*-*d*)/*a2*
end;
д) if *a*<*b* then begin *r*:=*a*; *a*:=*b*; *b*:=*r* end; {*a*>=*b*}
if *a*<*c* then begin *r*:=*a*; *a*:=*c*; *c*:=*r* end; {*a*>=*c*}
if *b*<*c* then begin *r*:=*b*; *b*:=*c*; *c*:=*r* end; {*b*>=*c*}
4.4. а) 1; б) 2; в) 0
4.5. if *w*<>0 then
begin if cos(*w*)/sin(*w*)<0.5 then *w*:=−*w* end
else *w*:=1

4.7. *b*:=*x*=*y*

4.8. if *a* then *x*:=true else

if *b* then *x*:=*c* else *x*:=false

4.11. Пустой оператор есть во всех примерах: а) между else и «;»; б) между then и else; в) между «;» и end; г) между первой и второй точками с запятой; д) между «1:» и end.

4.13. 6

4.14. а) 1-й вариант:

{при *a*>*b* НОД(*a*,*b*)=НОД(*a*−*b*, *b*)}
al:=*a*; *bl*:=*b*; {чтобы не портить *a* и *b*}

1: if *al*=*bl* then goto 2;

if *al*>*bl* then *al*:=*al*−*bl* else *bl*:=*bl*−*al*;

goto 1;

2: *c*:=*al*;

2-й вариант:

{при *a*>*b*>0 НОД(*a*,*b*)=НОД(*a* mod *b*,*b*); НОД(0,*b*)=*b*}

al:=*a*; *bl*:=*b*;

1: if (*al*=0) or (*bl*=0) then goto 2;

if *al*>*bl* then *al*:=*al* mod *bl* else *bl*:=*bl* mod *al*;

```

    goto 1;
2: if a1=0 then c:=b1 else c:=a1;
3) p:=1; i:=2;
9: p:=p*(l-1/sqr(i));
    i:=i+1; if i<=n then goto 9
4.18. program max (input,output);
    label 4,9;
    const n=50; {количество заданных чисел}
    var x,max:real; i:integer;
        {x—очередное число, i—его номер,
         max—наибольшее среди уже
         введенных чисел}
    begin read(max); i:=1;
        4: i:=i+1; if i>n then goto 9;
        read(x); if x>max then max:=x;
        goto 4;
        9: writeln('max=',max)
    end.

```

5. ОПЕРАТОР ЦИКЛА

5.1. {оператор цикла с предусловием:}

```

f:=1; f:=2; while i<=10 do begin f:=f+1;
                                i:=i+1 end;

```

{оператор цикла с постусловием:}

```

f:=1; i:=2; repeat f:=f+i; i:=i+1 until f>10;

```

{оператор цикла с параметром:}

```

f:=1; for i:=2 to 10 do f:=f+i

```

5.2. а) 0.2; б) 0.0; в) 1.0; г) 1.0

5.3. а) a1:=a; b1:=b; {чтобы не портить a и b}

```

        while (a1<>0) and (b1<>0) do

```

```

            if a1>b1 then a1:=a1 mod b1

```

```

            else b1:=b1 mod a1;

```

```

            if a1<>0 then c:=a1 else c:=b1;

```

б) n:=0; repeat n:=n+1; u:=-cos(cos(n)/sin(n))
 until u<0;

в) p:=1; for i:=2 to n do p:=p*(l-1/sqr(i));

г) y:=cos(40); for n:=39 downto 1 do y:=-cos(n+y)

5.4. x:=1; for n:=1 to k do x:=n*x+1/n

5.5. l:=0.001; r:=1.5; {левый и правый концы отрезка
 с корнем: tg(l)<1, tg(r)>1}

repeat x:=(l+r)/2; {середина отрезка [l,r]}

if sin(x)/cos(x)<x then l:=x {[l,r]:=[x,r]}

else r:=x {[l,r]:=[l,x]}

until r-l<1E-5;

$x := (l+r)/2$ {корень — середина последнего отрезка [l,r]}

5.9. program index (input,output);

var x,i,min,k:integer; {x и i — очередное число и его номер; min и k — минимальное из введенных чисел и его номер}

begin read(min); k:=1; read(x); i:=2;

while $x < > 0$ do

begin if $x < min$ then

begin min:=x; k:=i end;

read(x); i:=i+1

end;

writeln('номер минимального числа — ',k)

end.

5.11. a) $y := 1$; for $i := 2$ to 11 do $y := y * x + i$;

6) $y := 11$; for $i := 10$ downto 1 do $y := y * x + i$

5.13. a) $y := 1$; for $i := 1$ to n do $y := y * (2*i - 1)$

5.15. a) $y := 0$; $u := 1$; { $u = x^i = x * x^{i-1}$ }

for $i := 1$ to 30 do

begin $u := x * u$; $y := y + \cos(u)$ end;

6) $y := 1$; $f := 1$; { $u = f! = (i-1)! * i$ }

for $i := 2$ to n do begin $f := f * i$; $y := y + f$ end

5.16. a) $g := 1$; $f := 1$; { $g = f_0$, $f = f_1$ }

for $n := 2$ to 40 do

begin { $g = f_{n-2}$, $f = f_{n-1}$ }

$h := g$; $g := f$; $f := h + g$;

{ $g = f_{n-1}$, $f = f_n$ }

end

5.17. a) $y := 1$; {сумма} $n := 0$; {номер слагаемого}

$u := 1$; { n -е слагаемое $= x^n / n! = x^{n-1} / (n-1)! * (x/n)$ }

repeat $n := n + 1$; $u := u * x / n$; $y := y + u$ until $\text{abs}(u) < \text{eps}$

5.19. $y := 0$; for $i := 0$ to 10 do $y := y + \sin(1 + 0.1 * i)$

5.22. 3 раза (начальное и конечное значения параметра цикла вычисляются только раз — в начале выполнения оператора цикла).

5.23. $s := 0$;

for $k := \text{trunc}(\ln(x)) + 1$ to $\text{trunc}(\exp(x))$ do $s := s + \text{sqr}(k)$

5.25. a) $t := \text{true}$; $y := 1$; { $y = x^n = x * x^{n-1}$ }

for $n := 1$ to 30 do

begin $y := x * y$; if $\sin(y) < 0$ then goto 15 end;

$t := \text{false}$;

15: ...

6) $y := 1$; $n := 0$;

repeat $n := n + 1$; $y := x * y$; $t := \sin(y) < 0$ until t or ($n = 30$)

5.26. {У составного числа n обязательно есть делитель k из отрезка $[2, \sqrt{n}]$, он и ищется}

```
p:=true; k:=2;
while p and (sqr(k)<=n) do
  if n mod k=0 then p:=false else k:=k+1
```

5.29. a) p:=1;
 for i:=1 to 20 do
 for j:=1 to 20 do p:=p/(i+sqr(j))

5.30. k:=0;
{d1—левая, d2—средняя, d3—правая цифры числа}
for d1:=1 to 9 do
 for d2:=0 to 9 do
 for d3:=0 to 9 do
 if d1+d2+d3=n then k:=k+1

6. ЛИТЕРНЫЙ ТИП

6.1. Да: а), ж), в).

6.2. а) '6'; б) true; в) 5; г) true; д) true; е) true; ж) 23; з) '6'.

6.3. а) t:=d='*'; б) t:=(d='a')or(d='q');

в) t:=(d>='0')and(d<='9')

6.4. а) '+'; б) 'a'.

6.5. s:=ord('S')+ord('U')+ord('M')

6.6. writeln(chr(65),chr(71),chr(69))

6.7. If dig='9' then next:='0' else next:=succ(dig)

6.8. b:=ord('z')-ord('a')=25

6.9. for c:='A' to 'Z' do write(c); writeln

6.10. a) for d:='1' to '9' do

```
begin for c:='1' to pred(d) do write('0');
      write(d);
      for c:=succ(d) to '9' do write('0');
      writeln
end
```

6.11. program ab (input, output);

var c: char; {очередная линера текста}

ka, kb: integer; {число вхождений а и б}

begin ka:=0; kb:=0; read(c);

while c<>'.' do

```
begin if c='a' then ka:=ka+1 else
      if c='b' then kb:=kb+1;
      read(c)
end;
```

writeln(ka>kb)

end.

6.14. program целочисло (input,output);

var c:char; t:boolean;

begin t:=false;

```

read(c); if (c='+')or(c='—') then read(c);
if (c>='0')and(c<='9') then
begin repeat read(c)
      until (c<'0')or(c>'9');
      t:=c='.';
end;
if t then writeln ('правильно')
      else writeln ('неправильно')
end.

```

6.16.

- a) program безцифр (input,output);
var c:char;
begin read(c);
repeat if (c='+')or(c='—') then write(c,c) else
 if (c<'0')or(c>'9') then write(c);
 read(c)
until c='.';
writeln
end.
- b) program безлюса (input,output);
var a,b:char; {a — очередная литера текста,
 b — следующая}
begin read(a,b);
while b<>'.' do
begin if (a<>'+')or(b<'0')or(b>'9')
 then write(a);
 a:=b; read(b)
end;
writeln(a)
end.

6.19.

- a) program слова (Input,output);
var c:char; k:integer;
begin k:=0;
repeat {цикл по словам}
 read(c); {первая буква слова}
 if c='a' then k:=k+1;
 {пропуск остальных букв слова:}
 repeat read(c) until (c=',')or(c='.');
 until c='.'; {признак последнего слова}
 writeln ('с буквы "a" начинается ',k,' слов')
end.

6.20. n0:=ord('0');
k:=100*(ord(c2)-n0)+10*(ord(c1)-n0)+
 (ord(c0)-n0)

6.21. `n0:=ord('0');`
`d:=k div 100; c2:=chr(n0+d);`
`d:=k mod 100 div 10; c1:=chr(n0+d);`
`d:=k mod 10; c0:=chr(n0+d)`
 6.22. `{var c:char; sign, n0:integer;}`
 {определение знака числа:}
`sign:=1; read(c);`
`if c='-' then begin sing:=-1; read(c) end else`
`if c='+' then read(c);`
 {чтение цифр (первая — в с) и вычисление абсолютной величины числа k по схеме Горнера:}
`n0:=ord('0'); k:=0;`
`repeat k:=10*k+ord(c)-n0; read(c) until c=' ';`
 {учет знака:} `k:=sign*k`
 6.30. a) `for i:=1 to k-1 do write(' ');`
`writeln('*');`
 {или: `writeln('*':k)`}

b) `if k > n then max:=k else max:=n;`
`for i:=1 to max do`
`if i=k then write('*') else`
`if i=n then write('1') else write (' ');`
`writeln`

7. ПЕРЕЧИСЛИМЫЕ И ОГРАНИЧЕННЫЕ ТИПЫ. ОПЕРАТОР ВАРИАНТА

- 7.1. a) Недопустимы: 4), 5); 6) 1) true; 2) true; 3) false;
 4) ошибочное отношение; 5) лето; 6) зима; 7) ошибочное выражение; 8) тепло; а) 1; 2) 4; г) допустим; д) недопустимы: 1), 2), 3).
- 7.2. Правильно описаны типы *гласная*, *фигура*, *лог*.
- 7.3. а) Недопустимы: 2), 3); б) допустимы: 1), 3), 6); правильность 5) зависит от кодировки латер-цифр; в) ошибок нет; г) нельзя.
- 7.4. Правильно описаны типы *цифра*, *угол*, *плюс*, *неделя*, *будни*.
- 7.5. `t:=(m1 < m2)or(m1=m2)and(d1 < d2)`
- 7.6. а) `if m=дек then ml:=янв else ml:=succ(m);`
 б) `ml:=янв; for i:=1 to n-1 do ml:=succ(ml)`
- 7.11. а) `p=false, d=3;` б) `p=true, d=235;` в) `p=true, d=1;`
 г) ошибка.
- 7.13. `case m of`
`янв,фев,дек: s:=зима; мар,апр,май: s:=весна;`
`июн,июл,авг: s:=лето;сен,окт,ноя: s:=осень`
`end`

7.16. case с of

```
    черный: writeln('черный');
    серый: writeln('серый');
    белый: writeln('белый')
```

```
end
```

7.18. {var ch:char;}

```
read(ch);
```

```
case ch of
```

```
    'a': x:=a; 'b': x:=b; 'c': x:=c; 'd': x:=d
end
```

7.23. case m of

```
    янв,мар,май,июл,авг,окт,дек: d:=31;
```

```
    фев: d:=28;
```

```
    апр,июн,сен,ноя: d:=30
```

```
end
```

7.26. a) {var m1:месяц;}

```
{количество дней в предыдущих месяцах:}
```

```
    k:=0; m1:=янв;
```

```
    while m1 < m do
```

```
        begin case m1 of
```

```
            янв,мар,май,июл,авг,окт: k:=k+31;
```

```
            фев: k:=k+29;
```

```
            апр,июн,сен,ноя: k:=k+30
```

```
        end;
```

```
        m1:=succ(m1)
```

```
    end;
```

```
{учет дней в месяце m:} kt:=k+d
```

8. РЕГУЛЯРНЫЕ ТИПЫ: ВЕКТОРЫ

8.1. Массив a: а) 30; б) real; в) a[1], a[30].

Массив b: а) 5; б) x, y, z; в) b[-2], b[2].

Массив c: а) 10; б) массивы типа вектор; в) c['0'], c['9'].

Массив d: а) 3; б) целые от 0 до 23; в) d[чера], d[завтра].

8.2. type R=array [char] of 1..maxint

8.4. Неправильно описаны типы слово, вектор и цифры, а также переменная x.

8.5. а) y:=1; for i:=1 to n do y:=y*x[i];

```
        y:=exp(ln(abs(y))/n);
```

б) y:=x[1]; for i:=2 to n do if x[i]>y then y:=x[i];

в) y:=0; p:=1;

```
        for i:=1 to n do begin y:=y+p*x[i]; p:=-p end;
```

г) y:=0; for i:=1 to n do y:=y+x[i]*x[n+1-i];

д) y:=0; for i:=0 to n div 2 do y:=y+sqr(x[2*i+1]);

е) y:=1; p:=0;

```
        for i:=n downto 1 do begin p:=p+x[i]; y:=y*p end
```

- 8.7. a) {var max:integer; и1:имя;}
max:=139;
for и1:=Валя to Шура do
 if (Пол[и1]=муж) and (Рост[и1]>max) then
 begin max:=Рост[и1]; И:=и1 end
- 8.8. for I:=1 to 72 do t[i]:=k[t[i]]
- 8.12. I:=0;
repeat I:=I+1 until (I=n)or(a[I]<>b[I]);
if a[I]=b[I] then c:=a else
 for I:=1 to n do c[I]:=a[I]+b[I]
- 8.13. program ВводВыводМассива (input,output);
const n=100; d=6;
var x:array [1..n] of integer; I,k:integer;
begin {ввод массива:}
 for I:=1 to n do read(x[I]);
{вывод массива по d чисел в строке:}
 k:=0; {номер числа в строке}
 for I:=n downto 1 do
 begin k:=k+1; write(x[I]);
 if k=d then begin k:=0; writeln end
 end;
 if k>0 then writeln
 end.
- 8.14. for I:=0 to 39 do y[I]:=x[I+1]
- 8.15. Массивы не нужны в задачах а) и г).
- 8.16.
- a) program A (input,output); {с массивами}
const n=30; {четное число}
var x,y:array [1..n] of real;
I:integer; s1,s2:real;
begin for I:=1 to n do read(x[I]);
 for I:=1 to n do read(y[I]);
 s1:=0; {числитель} s2:=0; {знаменатель}
 for I:=1 to n div 2 do
 begin s1:=s1+x[2*I-1]*y[2*I-1];
 s2:=s2+x[2*I]*y[2*I]
 end;
 writeln(s1/s2)
- end.
- b) program Б (input,output); {без массивов}
const n=30;
var I:integer; x,y,s1,s2:real;
begin s1:=0; s2:=0;
for I:=1 to n do
 begin read(x,y); {ввод x[I] и y[I]}

```

        if odd(i) then s1:=s1+x*y else s2:=s2+x*y
    end;
    writeln(s1/s2)
end.

8.18. program СначалаЦифры (input,output);
    const n=80;
    var x:packed array [1..n] of char;
        c:char; i,k:integer;
begin k:=0; {количество нецифр}
for i:=1 to n do
begin read(c);
{цифру — на печать, нецифру — в массив x:}
if (c>='0')and(c<='9') then write(c) else
begin k:=k+1; x[k]:=c end
end;
{печатать всех литер из x:}
for i:=1 to k do write(x[i]); writeln
end.

8.21. a) p:=1; {очередная степень двойки}
repeat write(s[p]); p:=2*p until p>n;
writeln

8.26. 6) for k:=1 to n div 2 do
begin r:=x[k]; x[k]:=x[n+1-k];
x[n+1-k]:=r end;

r:=x[1]; for k:=1 to n-1 do x[k]:=x[k+1];
x[n]:=r

8.28. p:=1; {индекс очередного элемента из z}
i:=1; {из x} j:=1; {из y}
{пока есть нерассмотренные элементы и в x, и в y:}
repeat if x[i]<y[j] then begin z[p]:=x[i]; i:=i+1 end
else begin z[p]:=y[j]; j:=j+1 end;
p:=p+1
until (i>k)or(j>m);
{один из массивов x и y исчерпан → перепись в z оставшихся
элементов другого массива:}
if i>k {исчерпан массив x}
then repeat z[p]:=y[j]; j:=j+1; p:=p+1 until j>m
else repeat z[p]:=x[i]; i:=i+1; p:=p+1 until i>k

8.33. a) i:=1;
repeat t:=x[i]<x[i+1]; i:=i+1
until (not t)or(i=n)

8.35. {label 9; var ф1,ф2:фамилия;}
ЕстьТезки:=true;
for ф1:=Бетелии to pred(Школьный) do
for ф2:=succ(ф1) to Школьный do

```

```

    if MM5i1[φ1] == MM5i1[φ2] then goto 9;
ЕстьТезки:=false;
9; ...
8.38. a) for k:=n downto 2 do
begin {поиск m—номера max x[1..k]i}
  m:=1;
  for i:=2 to k do if x[i]>x[m] then m:=i;
  {перестановка элементов x[m] и x[k];}
  r:=x[k]; x[k]:=x[m]; x[m]:=r
end
8.39. k:=0; l:=1; r:=n;
{поиск p среди x[l..r];}
repeat m:=(l+r) div 2; {середина
                        отрезка l..r}
  if x[m]=p then k:=m else
    {замена l..r на m+1..r или l..m-1}
    if p>x[m] then l:=m+1 else r:=m-1
until (k<>0)or(l>r)
8.42. a) R[0]:=P[0];
      for i:=1 to n do R[i]:=R[i]-a*P[i-1];
      R[n+1]:=-a*P[n]

```

9. РЕГУЛЯРНЫЕ ТИПЫ; МАТРИЦЫ

9.1. Одннаковы.

9.3. Допустимые операции: а), д), е).

9.4. program ВводВыводМатрицы (input,output);

```

const n=4;
var A:array [1..n,1..n] of real; i,j:integer;
begin {ввод матрицы по строкам:}
  for i:=1 to n do
    for j:=1 to n do read(A[i,j]);
{вывод матрицы по столбцам:}
  for j:=1 to n do
    begin for i:=1 to n do write(A[i,j]);
      writeln
    end
  end,

```

9.5. a) for i:=1 to n do
 for j:=1 to n do C[i,j]:=A[i,j]+B[i,j];

б) $\{y[i] = \sum A[i,j]*x[j]\}$
for i:=1 to n do
begin s:=0;
 for j:=1 to n do s:=s+A[i,j]*x[j];
 y[i]:=s
end;

в) $\{C[i,j] = \sum A[i,k]*B[k,j]\}$
 for $i:=1$ to n do for $j:=1$ to n do
 begin $s:=0$;
 for $k:=1$ to n do $s:=s+A[i,k]*B[k,j]$;
 $C[i,j]:=s$
 end;
 г) $\{B[i,j] \leftrightarrow B[j,i], j > i\}$
 for $i:=1$ to $n-1$ do for $j:=i+1$ to n do
 begin $r:=B[i,j]; B[i,j]:=B[j,i]; B[j,i]:=r$ end

9.6. {label 1;}
 $t:=\text{false}$;
 for $i:=0$ to 9 do for $j:=-5$ to 3 do
 if $A[i,j] < > B[i,j]$ then goto 1;
 $t:=\text{true}$;
 1: ...

9.8. Неверны: а), в), е).

9.9. а) for $i:=0$ to 9 do $A[2*i+1]:=x$;
 б) for $j:=1$ to 10 do for $i:=1$ to 20 do $A[i,2*j]:=x[i]$;
 в) for $i:=1$ to 6 do for $j:=1$ to 20 do $B[i,j]:=x[j]$

9.11. $d:=0$;
 for $i:=1$ to 39 do
 for $j:=i+1$ to 40 do
 begin $r:=\text{sqr}(M[i,x]-M[j,x])+\text{sqr}(M[i,y]-M[j,y])$;
 if $r > d$ then $d:=r$
 end;
 $d:=\text{sqrt}(d)$

9.13. {var il:остров; m1:месяц;}
 $il:=\text{Барбадос}; m1:=\text{янв};$
 for $il:=\text{Барбадос}$ to Ямайка do
 for $m1:=\text{янв}$ to дек do
 if $t[il,m1] > t[i,m]$ then begin $i:=il; m:=m1$ end

9.14. а) $s:=A[1,1]+A[9,1]+A[1,9]+A[9,9]$
 for $i:=2$ to 8 do
 $s:=s+A[1,i]+A[9,i]+A[i,1]+A[i,9]$

9.16. а) for $i:=1$ to 10 do $A[i,i]:=0$;
 for $i:=2$ to 10 do
 begin $A[i]:=A[1]; A[i,i]:=i-1$ end

9.20. а) for $k:=1$ to 15 do
 begin $j:=0$;
 repeat $j:=j+1$ until $(A[k,j] < > 0) \text{ or } (j=20)$;
 $b[k]:=A[k,j]=0$
 end

9.23.
 а) program sort (input,output);
 const n=20; m=30;

```

type строка=array [1..m] of real;
матрица=array [1..n] of строка;
var A:матрица; x:строка; i,j,k:integer;
begin {ввод A:}
  for i:=1 to n do for j:=1 to m do read(A[i,j]);
{сортировка выбором (см. 8.38,а:)}
  for k:=n downto 2 do
    begin {поиск j—номера max A[1..k,1]:}
      j:=1;
      for l:=2 to k do if A[l,1]>A[j,1] then j:=l;
{перестановка k-й и j-й строк:}
      x:=A[k]; A[k]:=A[j]; A[j]:=x
    end;
{вывод A:}
  for i:=1 to n do
    begin writeln (i,'-я строка:');
      for j:=1 to m do write(A[i,j]); writeln
    end
  end.

```

9.25. a) {label 1; var Р,ДР:имя;}

{поиск Р—ребенка И, а затем поиск ДР—дочери Р}

for Р:=Анна to Юрий do

 if (TP[И,Р]=сын)or(TP[И,Р]=дочь) then

 for ДР:=Анна to Юрий do

 if TP[Р,ДР]=дочь then begin B:=ДР; goto 1 end;

1: ...

10. РЕГУЛЯРНЫЕ ТИПЫ; СТРОКИ

10.1. Строковым является только тип b.

10.2. Допустимые операции: а), б), г), д), з), к), и), о).

10.3. програм ВводВыводСтроки(input,output);

```

const n=60;
var s:packed array [1..n] of char; l:integer;
begin
  for i:=1 to n do read(s[i]);
  writeln(s); writeln(s)
end.

```

10.4. {обработка массива a1:}

k1:=0;

for i:=2 to 20 do

begin j:=0;

repeat j:=j+1 until (a1[i,j]<>a1[1,j])or(j=8);

if a1[i,j]=a1[1,j] then k1:=k1+1

end;

{обработка массива a2:}

k2:=0;
for i:=2 to 20 do if a2[i]=a2[1] then k2:=k2+1

10.5. a) for i:=1 to 60 do
 if C[i]<>'hello' then write(C[i], ' ');
 writeln;

b) for i:=1 to 60 do write(C[i,5]); writeln

10.6. {var c:char; i:integer;}

{сначала запись в s шести пробелов, а затем замена первых пробелов введенными буквами:}

s:=' '; i:=0; read(c);

while c<>'.' do begin i:=i+1; s[i]:=c; read(c) end

10.8. {type string=packed array [1..6] of char;

var n:array [0..5] of string;

 y:string; i:integer; c:char; ok:boolean;}

{формирование массива названий цветов в виде строк:}

n[0]:='red '; n[1]:='blue '; n[2]:='green ';

n[3]:='yellow'; n[4]:='black '; n[5]:='white ';

{ввод не более 6 букв до пробела и запись их в y:}

y:=' '; i:=0; read(c);

while (c<>' ')and(i<6) do

 begin i:=i+1; y[i]:=c; read(c) end;

ok:=(i>0)and(c=' '); {от 1 до 6 букв и пробел
за ними?}

if ok then

 begin {если y=n[i], то x:=i-е имя типа color:}

 i:=-1; repeat i:=i+1 until (y=n[i])or(i=5);

 if y<>n[i] then ok:=false else

 begin x:=red;

 for i:=0 to i-1 do x:=succ(x) end

 end;

 if not ok then writeln ('неверное значение типа color')

10.10. {var v1: packed array [1..5] of char;

 i,j:integer;}

 k:=0; v1:=v; {копирование v в v1,

иначе v[i]—ошибка}

for i:=1 to 200 do

 begin j:=0; repeat j:=j+1 until (s[i]=v1[j])or(j=5);

 if s[i]=v1[j] then k:=k+1

 end

10.11. program русскиебуквы (input,output);

 const n=60;

 var p:packed array [1..32] of char;

 c:char; i,j:integer;

 begin p:='абвгдежзийклмиопстуфхцчшъыъюя';

 for i:=1 to n do

```

begin read(c); j:=0;
repeat j:=j+1
until (c=p[j])or(j=32);
if c=p[j] then write(c)
end;
writeln
end.

10.15. 6) {поиск в s буквы 'w' или пробела:}
i:=1; while (s[i]<>'w')and(s[i]<>' ') do i:=i+1;
if s[i]='w' then
{сдвиг следующих за 'w' букв на 1 позицию влево и запись
пробела вместо последней буквы:}
repeat i:=i+1; s[i-1]:=s[i] until s[i]=' ';
g) {поиск в s буквы 'x' или пробела:}
i:=1; while (s[i]<>'x')and(s[i]<>' ') do i:=i+1;
if s[i]='x' then
begin s[i]:='k'; {замена 'x' на 'k'}
{вставка 's' за 'k' и сдвиг последующих букв вправо:}
b:='s'; {вставляемая литература}
repeat i:=i+1; {куда вставлять}
c:=s[i]; {освободить место для b}
s[i]:=b; {вставка b}
b:=c {новая вставляемая литература}
until b=' '
end

```

11. ФУНКЦИИ И ПРОЦЕДУРЫ

```

11.1. 6) function sign(a:real):integer;
begin if a<0 then sign:=-1 else
      if a>0 then sign:=1 else sign:=0
end;
...
z:=(sign(x)+sign(y))*sign(x+y)
11.2. program shtg (input,output);
var x,y:real;
function sh(z:real):real;
var e:real;
begin e:=exp(z); sh:=(e-1/e)/2 end;
function tg(z:real):real;
begin tg:=sin(z)/cos(z) end;
begin read(x);
y:=sh(x)*tg(x+1)-sqr(tg(2+sh(x-1)));
writeln(y)
end.

```

```

11.3. function конт(s:страна):континент;
begin case s of
    Австрия,Италия,Швеция: конт:=Европа;
    Гана: конт:=Африка;
    Перу,США: конт:=Америка
end
end;
...
if конт(x)<>конт(y) then t:=not t
11.4. function степень(x:real; n:натур):real;
var i:integer; y:real;
begin y:=1; for i:=1 to n do y:=y*x;
степень:=y
end;
...
b:=степень(2.7,k)+1/степень(a+1,5)
11.6. a) 2 8 6
11.8. program площадь (input,output);
var a,b,c,d:real;
procedure печплощ(x,y,z:real);
var p:real;
begin if (x+y>z)and(y+z>x)
        and(z+x>y) then
begin p:=(x+y+z)/2;
writeln(sqrt(p*(p-x)*
                (p-y)*(p-z)))
end
end;
begin read(a,b,c,d); {a,b,c,d>0}
    печплощ(a,b,c); печплощ(a,b,d);
    печплощ(a,c,d); печплощ(b,c,d)
end.
11.9. б) 0 и 7; в) обращение  $R(\text{sqr}(c)+c,d)$  недопустимо.
11.10. б) Оба обращения недопустимы;
в) procedure сокр(a,b:integer; var p,q:integer);
var a1,b1,f:integer;
begin a1:=abs(a); b1:=abs(b);
while (a1<>0)and(b1<>0) do
    if a1>b1 then a1:=a1 mod b1
    else b1:=b1 mod a1;
    if a1=0 then f:=b1 else f:=a1;
    p:=a div f; q:=b div f
end;
...
c:=0; d:=1; {c/d=0}

```

```
for i:=1 to 20 do
    скр(c*i+d,i*d,c,d) {c/d+1/i=(ci+d)/(di)}
```

11.11. а) Оба обращения недопустимы;

в) procedure maxmin(var x,y:real);
 var r:real;
begin if x<y then begin r:=x; x:=y; y:=r end end;
...
maxmin(a,b); maxmin(a,c); {a=max}
maxmin(b,c); {c=min}

11.12. б) procedure sum(var x,y,z:вектор);

```
    var i:integer;
begin for i:=1 to n do z[i]:=x[i]+y[i] end;
...
sum(a,b,d); sum(d,c,d)
```

11.18.

```
program многочлены (input,output);
const n=30;
type вект=array [0..n] of real;
var a,b,c:вект; x,y,z,d:real;
procedure ввод(var v:вект); {ввод вектора}
    var i:integer;
begin for i:=0 to n do read(v[i]) end;
function знач(var v:вект; t:real):real;
{значение многочлена с коэффициентами из v в
точке t}
    var s:real; i:integer;
begin s:=v[0]; for i:=1 to n do s:=s*t+v[i];
    знач:=s
end;
begin
    ввод(a); ввод(b); ввод(c); read(x,y,z);
    d:=(sqrt(знач(a,x))-знач(b,y))/знач(c,x+z);
    writeln(d)
end.
```

11.24. 0.0 2.0

11.25. procedure сдв(var s:шкала; k:сдвиг);
 var i:integer; t:шкала; {вспомогательный
 массив}
begin {t[n-k+1..n]:=s[1..k]:}
 for i:=1 to k do t[n-k+i]:=s[i];
 {t[1..n-k]:=s[k+1..n]:}
 for i:=k+1 to n do t[i-k]:=s[i];
 s:=t
end

11.27. function F(m,n:неотриц):real;

```

        function fact(k:неотриц):неотриц;
          var l,p:integer;
          begin p:=1; for i:=2 to k do p:=p*i;
              fact:=p end;
          begin F:=fact(n)*fact(m)/fact(n+m) end
11.28. procedure преобр(var x,y,a,b:вектор);
    procedure вставка(var w,v:вектор; n,d:integer);
      {w[n..n+d-1]:=v[1..d]}
      var i:integer;
      begin for i:=1 to d do w[n-1+i]:=v[i] end;
      begin вставка(x,a,1,8);
          вставка(y,b,6,6);
          вставка(y,a,17,4)
      end
11.32. Локальные: x, y, z, с; глобальные: vect, integer, index, a, b.
11.33. a) 8 true; a 5; 6 *; a * б) 9 в) 6 6 7 8; 6 2 3 4
11.40.
program integrals (input,output);
var c,d:real;
function f1(x:real):real;
begin f1:=sqr(arctan(x)) end;
function f2(x:real):real;
begin f2:=sin(exp(10*x)) end;
function int(function f(x:real):real; a,b:real;
            n:integer):real;
var h,s:real; i:integer;
begin h:=(b-a)/n; s:=(f(a)+f(b))/2;
   for i:=1 to n-1 do s:=s+f(a+i*h);
   int:=h*s;
end; {of int}
begin read(c,d);
writeln('int(f1,c,d,20)+int(f2,0,3.1415927,100)')
end.
11.44. 2 1 false
11.46. function next:char;
        var c:char;
        begin repeat read(c) until c<>'.'; next:=c end;
        ***
k:=0; while next<>'.,' do k:=k+1

```

12. РЕКУРСИЯ

- 12.1. fib(2)=2; fib(4)=5.
 12.2. Правильное: r).
 12.3. function pow(x:real; n:integer):real;

```

begin if n=0 then pow:=1 else
    if n<0 then pow:=1/pow(x,abs(n))
        else pow:=x*pow(x,n-1)
end

```

12.4. function C(m,n:integer):integer;

```

begin if (m=0)or(m=n) then C:=1
        else C:=C(m,n-1)+C(m-1,n-1)
end

```

12.5.

```

function Потомок(a,b:имя):boolean;
var f,м:имя;
begin if (a=b)or(a=нет)or(b=нет)
then Потомок:=false
else
begin f:=Отец(b); m:=Мать(b);
if (a=f)or(a=m)
then Потомок:=true else
{ребенок потомка—также потомок}
if Потомок(a,f) then Потомок:=true
else Потомок:=Потомок(a,m)
end
end

```

12.7. $f(n)=\begin{cases} n-10 & \text{при } n>100, \\ 91 & \text{иначе.} \end{cases}$

12.9.

```

function min(var x:вектор):real;
function min1(k:integer):real; {min x[k..n]}
var m:real;
begin if k=n then min1:=x[n] {min x[n..n]=x[n]}
else
begin {min x[k..n]=min(x[k],
min x[k+1..n])}
m:=min1(k+1);
if x[k]<m then min1:=x[k]
else min1:=m
end
end; {of min1}
begin min:=min1(1) end

```

12.11. function sum:real;

```

var x:real;
begin read(x); {1-е число}
if x<0 then sum:=0 else
{сумма всех чисел=1-е число+сумма
остальных}

```

```

        sum:=x+sum
    end
12.15.
program formula (input,output);
functionF:integer;
{F читает из начала входного файла литеры, образующие
законченную формулу, и вычисляет ее значение как целое}
var c,op:char; x,y:integer;
begin read(c);
if (c>='0')and(c<='9')
then {цифра есть формула} F:=ord(c)-ord('0')
else {началась формула вида (x op y)}
begin x:=F; read(op); y:=F;
case op of
  '+': F:=x+y; '-': F:=x-y; '*': F:=x*y
end;
read(c) {пропуск ')'}
end
end; {of F}
begin writeln(F) end.

```

13. КОМБИНИРОВАННЫЕ ТИПЫ. ОПЕРАТОР ПРИСОЕДИНЕНИЯ

13.1. б) type время=record час:0..23;
мин,сек:0..59 end;

ж) type строка=packed array [1..12] of char;
ведомость=
record предмет:строка;
номергруппы:integer;
дата:record ч:1..31; м:1..12;
г:integer end;
студенты:array [1..25] of
record фамилия:строка;
номерзачетки:integer;
оценка:2..5
end
end

13.2. function бьет(var K1,K2:карта;
КМ:масть):boolean;
begin if K1.м=K2.м then бьет:=K1.д>K2.д
else бьет:=K1.м=КМ
end

13.3. procedure СамаяВысокая(var С:список);
var m,i:integer;
begin m:=1;

```
for i:=2 to 30 do
    if C[i].высота > C[m].высота then m:=i;
    writeln(C[m].название)
end
```

13.6. б) $d := \sqrt{(p1[x] - p2.x)^2 + (p1[y] - p2.y)^2}$

13.7. with Adr1 do
begin город:='Москва'; улица:='Арбат'
дом:=1; квартира:=5
end;

Адр2:=Адр1; Адр2.квартира:=17

13.8. Правильно: б), в), г).

13.10. z.re=z.im=1; w.re=0, w.im=-1;
p.x=p.y=2; re=2

13.14.

6) procedure следсек(var t,t1:время);

```
begin t1:=t;
with t1 do
if сек<59 then сек:=сек+1 else
begin сек:=0;
if мин<59 then мин:=мин+1 else
begin мин:=0; час:=(час+1) mod 24 end
end
end
```

13.16.

a) function равно(var a,b:раз):boolean;
begin равно:=a.числ*b.знам=b.числ*a.знам end

14. МНОЖЕСТВЕННЫЕ ТИПЫ

14.1. б) Возможные значения переменных!

х: [], [0], [1], [0,1];

у: [], [a], [b], [c], [a,b], [a,c], [b,c], [a,b,c];

z: [], ['*'].

14.2. 2^n .

14.3. type A=set of день недели; Б=set of пн..пт;

14.4. Неправильно определены типы точки, данные, М3, М4.

14.5. Не множества: ж), з), и).

14.6. а) [2,5,6]; б) []; в) ошибка.

14.8. а), г), е), к)—false; б), в), д), ж), з), и)—true; л), м)—ошибка.

14.10. function счет(var s:строка):integer;
var i,k:integer;
begin k:=0;
for i:=1 to 100 do
if s[i] in ['0'..'9','+', '-','*'] then k:=k+1;

```

        счет:=к
    end
14.13. function card(A:M):integer;
    var p,k:integer;
begin k:=0;
    for p:=0 to 99 do if p in A then k:=k+1;
    card:=k
end
14.14. procedure print(A:letters);
    var c:char;
begin for c:='a' to 'z' do
    if c in A then write(c);
    writeln
end
14.16. Правильно: г), д).
14.17. а) [1..5]; б) [ ]; в) [1,3,5]; г) [1..8]; д) [3..6]; е) [1,2];
ж) [1..5]; з) [2,4]; и) [ ]; к) [4]; л) [ ]; м) [ ].
14.18. а) [3..5,7..10,13]
14.19. а) [ ]; б) А*B.
14.21. x:=[8..22]; y:=[11,13,17,19]; z:=x-y
14.23. а) B:=A+[x]; б) B:=A-[x]
14.24.
а) function digits(n:натур):натур;
    var sd:set of 0..9; d:0..9; k:integer;
begin {выделение справа налево цифр из n запись их в множество sd:}
    sd:=[ ];
    repeat d:=n mod 10; sd:=sd+[d];
    n:=n div 10
    until n=0;
    {подсчет числа элементов в sd:}
    k:=0;
    for d:=0 to 9 do if d in sd then k:=k+1;
    digits:=k
end
14.25.
а) program первыевхождения (input,output);
    var let:set of 'a'..'z'; c:char;
begin let:=[ ]; {множество букв в рассмотренной части текста}
    read(c);
    while c<>'.' do
        begin if not(c in let) then {1-е вхождение}
            begin write(c); let:=let+[c] end;
            read(c)
        end
end

```

```
    end;  
    writeln  
end.
```

15. ФАЙЛОВЫЕ ТИПЫ

15.2. а) 7; б) 10; в) 21

```
15.4. function отриц(var s:серия):real;  
    var sum,x:real;  
begin reset(s); sum:=0;  
    while not eof(s) do  
        begin read(s,x);  
            if x<0 then sum:=sum+x end;  
        отриц:=sum  
    end
```

```
15.6. function унор(var r:ряд):boolean;  
    var x,y:0..999; ok:boolean;  
begin reset(r); read(r,y); ok:=true;  
    while not eof(r) and ok do  
        begin x:=y; read(r,y); ok:=x<y end;  
    унор:=ok  
end
```

```
15.7. function eq(var t1,t2:текст):boolean;  
    var c1,c2:char; ok:boolean;  
begin reset(t1); reset(t2); ok:=true;  
    while not eof(t1) and not eof(t2) and ok do  
        begin read(t1,c1);  
            read(t2,c2); ok:=c1=c2 end;  
        eq:=ok and eof(t1) and eof(t2)  
    end
```

15.11. а) 1, 3; б) 9, 4, 1

```
15.12. procedure цифры(var s:строка; var t:текст);  
    var i:integer;  
begin rewrite(t);  
    for i:=1 to 100 do  
        if s[i] in ['0'..'9'] then write(t,s[i])  
    end
```

```
15.14. procedure присв(var f,g:FB);  
    var b:boolean;  
begin reset(g); rewrite(f);  
    while not eof(g) do  
        begin read(g,b); write(f,b) end  
    end
```

```
15.17. function less(var f:reals):integer;  
    var k:integer; x,s:real;  
begin {подсчет среднего арифметического:}
```

- ```

reset(f); k:=0; s:=0;
repeat read(f,x); k:=k+1;
 s:=s+x until eof(f);
s:=s/k;
{новый просмотр f и подсчет элементов <s:}
reset(f); k:=0;
repeat read(f,x);
 if x < s then k:=k+1
until eof(f);
less:=k
end
15.21. a) procedure add1(var t:текст; c:char);
 var d:char; s:текст; {вспомогательный
 файл}
 begin {копирование t в s:}
 reset(t); rewrite(s);
 while not eof(t) do
 begin read(t,d); write(s,d) end;
 {запись с и s в t:}
 reset(s); rewrite(t);
 write(t,c);
 while not eof(s) do
 begin read(s,d); write(t,d) end
 end
15.22. а) 2; б) 4; в) ошибка; г) 2
15.23. а) 'a'; б) ошибка; в) '1', '2', '3'; г) '1', '3', '2'.
15.25. procedure присв(var f,g:Ф);
 begin reset(g); rewrite(f);
 while not eof(g) do
 begin write(f,g↑); get(g) end
 end
15.26. function mid(var f:FR; var m:real):boolean;
 var k,i:integer;
 begin {подсчет k — числа элементов в f:}
 reset(f); k:=0;
 while not eof(f) do
 begin k:=k+1; get(f) end;
 mid:=odd(k);
 if odd(k) then {поиск среднего элемента}
 begin reset(f);
 for i:=1 to k div 2 do get(f); m:=f↑
 end
 end
15.31. procedure triangle(var t:text);
 var c,d:char;

```

- ```

begin rewrite(t);
  for d:= '1' to '9' do
    begin for c:='1' to d do write(t,d);
      writeln(t)
    end
  end

```
- 15.33. a) function empty(var t:text):integer;
 var k,d:integer;
 begin reset(t); k:=0;
 while not eof(t) do {цикл по строкам}
 begin d:=0; {d—длина очередной
 строки}
 while not eoin(t) do
 begin d:=d+1; get(t) end;
 if d=0 then k:=k+1;
 get(t) {пропуск «конца строки»}
 end;
 empty:=k
 end
- 15.35. a) function count(var t:text):integer;
 var k:integer; c:char;
 begin reset(t); k:=0;
 while not eof(t) do
 begin {учет первой буквы строки:}
 read(t,c); if c='d' then k:=k+1;
 readln(t) {пропуск остатка
 строки}
 end;
 count:=k
 end
- 15.36. procedure присв(var t1,t2:text);
begin reset(t2); rewrite(t1);
 while not eof(t2) do
 if eoin(t2)
 then begin writeln(t1); get(t2) end
 else begin write(t1,t2); get(t2) end
 end
- 15.39. procedure сан(var l:список; var t:text);
 var i:integer;
 begin rewrite(t);
 for i:=1 to 100 do writeln(t,i[i])
 end
- 15.40. function max(var t:text):real;
 var m,x:real;
 begin reset(t); read(t,m);

```

        while not eof(t) do
            begin read(t,x); if x>m then m:=x end;
            max:=m
        end
15.42. with d do
        writeln (число:1,'.',месяц:1,'.',год:4)
15.47. program formatting(BOOK);
    const d=60; {длина «новых» строк}
    var BOOK,COPY:text; c:char; k:integer;
begin {перепись BOOK в COPY с удалением «коцов
строк»:}
    reset(BOOK); rewrite(COPY);
    while not eof(BOOK) do
        if eoln(BOOK) then get(BOOK) else
            begin read(BOOK,c); write(COPY,c) end;
        {перенос текста в BOOK с форматированием:}
    reset(COPY); rewrite(BOOK);
    k:=0; {k — порядковый номер литеры в строке}
    while not eof(COPY) do
        begin read(COPY,c); write(BOOK,c); k:=k+1;
        if (c='.')or(k=d) then
            begin writeln(BOOK); k:=0 end
        end
    end.

```

16. ССЫЛОЧНЫЕ ТИПЫ. СПИСКИ

16.1. б) 4

16.2. см. рис. 24.

16.3. Неправильные: б), д), е), ж), и), к), л), м).

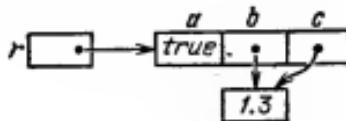


Рис. 24

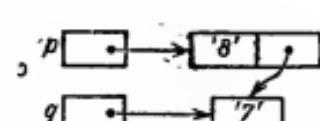


Рис. 25

16.5. см. рис. 25.

16.6. г) см. рис. 26.

16.8. type D=0..9; A=†D;

C=†B; B=record p:real; q:C end;

16.9. б) type C=†char; A=†R;

R=record f1:C; f2:A end;

var p,q:A; x:C;

...

```

new(p); new(q);
new(x); x^.f1:='a'; p^.f1:=x; p^.f2:=q;
new(x); x^.f1:='b'; q^.f1:=x; q^.f2:=nil

```

16.10. 1) $q := p$; $p := p^.след$; $\text{dispose}(q)$

16.12. б) function negl (var x:вектор):ссылка;
 var i:integer; t:boolean;
 begin i:=0;
 repeat i:=i+1; t:=x[i]<0
 until t or (i=100);
 if t then negl:=x[i] else negl:=nil
 end

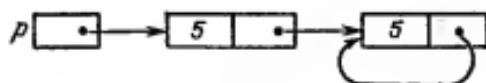


Рис. 26

16.14. в) {список без заглавного звена}

```

procedure замена (L:список; E1,E2:ТЭ);
var p:список; {ссылка на очередное звено}
begin p:=L;
while p<>nil do
begin if p^.элем=E1 then p^.элем:=E2;
      p:=p^.след {переход к следующему звено}
end
end;

```

д) {список с заглавным звеном}

```

function упор(L:список):boolean;
var p,q:список; {ссылки на пару
                  соседних звеньев}
ok:boolean;
begin ok:=true;
p:=L^.след; {nil или ссылка на 1-зено}
if p<>nil then
begin q:=p^.след; {nil или ссылка на
                    2-е звено}
while (q<>nil) and ok do
begin ok:=p^.элем<=q^.элем;
   p:=q; q:=q^.след {переход к след. паре}
end
end;
упор:=ok
end

```

16.16. а) {список с заглавным звеном}

```
function newlist(var f:файл):список;
```

```

var L,p,q:список; x:ТЭ;
begin reset(f);
  {создать заглавное звено:} new(L);
  p:=L; {ссылка на последнее из созданных
         звеньев}
  while not eof(f) do
    begin read(f,x);
      {создать звено с x:} new(q); q^.элем:=x;
      {добавить его за p:} p^.след:=q;
      {объявить его последним:} p:=q
    end;
  p^.след:=nil; {записать nil в последнее звено}
  newlist:=L
end

```

16.18. д) {список без заглавного звена}

```

procedure вставка(L:список; E,E1:ТЭ);
  var p,q:список; eq:boolean;
begin { поиск звена с E:}
  p:=L; eq:=false;
  while (p<>nil) and not eq do
    if p^.элем=E then eq:=true
    else p:=p^.след;
  if eq then {вставка E1 перед E}
    begin {внимание—трюк: запись E1 в звено p
           вместо E и вставка E за звеном p:}
      p^.элем:=E1; new(q); q^.элем:=E;
      q^.след:=p^.след; p^.след:=q
    end
  end

```

16.19. г) {список с заглавным звеном}

```

procedure удалить(L:список);
  var p,q:список;
begin { поиск предпослед. (p) и послед. (q) звеньев:}
  p:=L; q:=p^.след;
  while q^.след<>nil do begin p:=q; q:=q^.след end;
  {удаление последнего звена:}
  dispose(q); p^.след:=nil
end

```

16.20. program reverse (input,output);

```

type список=^звено;
  звено=record элем:char; след:список end;
  var L,p:список; c:char;
begin {ввод литер текста и запись их в обратном порядке
       список L (без заглавного звена):}
  L:=nil; {ссылка на построенную часть списка}

```

```
read(c);
while c<>'.' do
begin {добавить с в начало списка:}
    new(p); p^.элем:=c;
    p^.след:=L; L:=p; read(c)
end;
{печатать литер из L:}
while L<>nil do
begin write(L^.элем); L:=L^.след end;
writeln
end.
```

16.23. e) {списки с заглавными звеньями}

```
procedure append(L1,L2:список);
var p1:список;
begin {поиск конца списка L1:}
p1:=L1; while p1^.след<>nil do p1:=p1^.след;
{вставка в конец L1 ссылки на 1-е звено L2:}
p1^.след:=L2^.след
end
```

16.24. {списки без заглавных звеньев}

- a) function memb(L:список; E:TЭ): boolean;
begin if L=nil then memb:=false else
 if L^.элем=E then memb:=true
 else memb:=memb(L^.след,E)
end;
- e) procedure delete(var L:список; E:TЭ);
var p:список;
begin if L<>nil then
 if L^.элем=E then {удалить 1-е звено}
 begin p:=L; L:=L^.след; dispose(p) end
 else {удалить Е из «хвоста» списка
 и записать в 1-е звено ссылку на измененный «хвост:}
 delete(L^.след,E)
end

16.29. б) {списки без заглавных звеньев}

```
procedure firstletters(L:список);
begin while L<>nil do
begin if L^.элем<>nil
        then write(L^.элем^.буква);
        L:=L^.след
end;
writeln
```

- end
- 16.33. a) function empty(L:список2):boolean;
begin empty:=L^.след=L end;

6) procedure reprint(L:список2);
var p:список2;
begin p:=L.пред; {последнее звено}
while p<>L do begin write(p^.элем); p:=p^.пред end;
writeln
end

17. ОЧЕРЕДИ, СТЕКИ, ДВОИЧНЫЕ ДЕРЕВЬЯ

```

17.1. a) type очередь=record нач,кон:0..п;
                         мас:array [1..п] of ТЭО
                     end;
procedure ОЧИСТОЧ(var Q:очередь);
begin Q.нач:=1; Q.кон:=0 end;
function ПУСТОЧ(var Q:очередь):boolean;
begin ПУСТОЧ:=Q.кон=0 end;
procedure ВОЧЕРЕДЬ(var Q:очередь; x:TЭО);
var i:integer;
begin with Q do
begin if кон=п then {очередь у правого края}
           If нач=1 then {занят весь мас} ОШИБКА(1)
           else {сдвиг к левому краю}
                 for i:=нач to кон do
                   мас[i-нач+1]:=мас[i];
           кон:=кон+1; мас[кон]:=x
end
end;
procedure ИЗОЧЕРЕДИ(var Q:очередь; var x:TЭО);
begin if ПУСТОЧ(Q) then ОШИБКА(2);
       with Q do
begin x:=мас[нач];
       if нач=кон then {был 1 элемент}
                     ОЧИСТОЧ(Q)
       else нач:=нач+1
end
end;

```

17.2.

```

a) procedure печать(var f:FR; a,b:real);
    var Q1,Q2:очередь; {ТЭО=real} x:real;
begin reset(f); ОЧИСТОЧ(Q1); ОЧИСТОЧ(Q2);
{печатать чисел <a, запись в Q1 чисел ∈[a,b] и
запись в Q2 чисел >b:}
    while not eof(f) do
        begin read(f,x);
            if x<a then write(x) else

```

```

        if x<=b then ВОЧЕРЕДЬ(Q1,x)
        else ВОЧЕРЕДЬ(Q2,x)
    end;
{печатать чисел из Q1 и Q2:}
    while not ПУСТОЧ(Q1) do
        begin ИЗОЧЕРЕДИ(Q1,x); writeln(x) end;
    while not ПУСТОЧ(Q2) do
        begin ИЗОЧЕРЕДИ(Q2,x); write(x) end;
    writeln
end

```

17.4.

b) function formula(var t:text):integer;
var S:стек; c,op,x,y:char;
begin reset(t); ОЧИСТЕК(S);
while not eof(t) do
begin read(t,c);
{обработка очередной литеры текста (литеры '('
и ',' игнорируются):}
if c in ['0'..'9','M','m'] then ВСТЕК(S,c) else
if c=')' then {конец формулы вида op(x,y)}
begin {в конце стека находится тройка op x y,
она удаляется из стекла, выполняется
операция op и результат записывается
в стек:}
 ИЗСТЕКА(S,y); ИЗСТЕКА(S,x); ИЗСТЕКА(S,op);
case op of
 'M' {max}: if x>y then c:=x
 else c:=y;
 'm' {min}: if x<y then c:=x else c:=y
end;
 ВСТЕК(S,c)
end
end; {of while}
{в стеке осталась одна цифра—значение всей
формулы; цифра переводится в целое число:}
 ИЗСТЕКА(S,c); formula:=ord(c)-ord('0')
end

17.7.

6) function count(T:дерево; E:TЭД):integer;
var S:стек; k:integer;
begin ОЧИСТЕК(S); k:=0; {число вершин с E}
while T<>nil do
begin {T—ссылка на очередную вершину}
if T^.элем=E then k:=k+1;
{переход к следующей вершине:}

```

if T^.лев <> nil then {есть ветвь влево}
begin {правую ветвь, если есть,— в стек:}
      if T^.прав <> nil then ВСТЕК(S,T^.прав);
      {идти влево;} T:=T^.лев
end
else if T^.прав <> nil then {идти вправо:}
      T:=T^.прав
else {нет обеих ветвей}
begin {взять ветвь из стека и идти по ней:}
      if ПУСТЕК(S) then T:=nil {конец
                                     просмотр}
      else ИЗСТЕКА(S,T)
end
end; {of while}
count:=k
end;
ж) procedure levels (T: дерево);
var Q: очередь; n,k,k1:integer;
begin
  if T<>nil then
    begin ОЧИСТОЧ (Q);
    n:=0; ВОЧЕРЕДЬ (Q,T); k:=1;
    repeat {цикл по уровням}
      {В очереди Q находятся все (k) вершины (ссылки на
       них) n-го уровня дерева. Оничитываются и печатаются, и одновременно в конец Q заносятся все (k1) их дочерние вершины, т. е. вершины (n+1)-го уровня.}
      write (n, '-й уровень:');
      k1:=0;
      for k:=1 to k do
        begin ИЗОЧЕРЕДИ(Q,T);
        write (' ',T^.элем);
        if T^.лев <> nil then
          begin ВОЧЕРЕДЬ(Q,T^.лев); k1:=k1+1 end;
        if T^.прав <> nil then
          begin ВОЧЕРЕДЬ (Q,T^.прав); k1:=k1+1 end
        end;
      writeln; {конец печати n-го уровня}
      n:=n+1; k:=k1 {на следующий уровень}
    until k=0 .
  end
end
17.8. б) function count (T:дерево; E:TЭД):integer;
var k:integer;

```

```

begin
  if T=nil then count:=0 else
    begin if T^.элем=E then k:=1 else k:=0;
           count:=k+count(T^.лев,E)+  

           count(T^.прав,E)
    end
  end;

```

e) function height(T:дерево): integer;

var h1,h2:integer;

begin

```

  if T=nil then height:=-1 else
    begin h1:=height(T^.лев);
          h2:=height(T^.прав);
          if h1>h2 then height:=h1+1
          else height:=h2+1
    end
end

```

17.10. procedure copy(T:дерево; var T1:дерево);
begin if T=nil then T1:=nil else
begin new(T1); T1^.элем:=T^.элем;
copy(T^.лев,T1^.лев);
copy(T^.прав,T1^.прав)
end
end

17.14

a) procedure освоб(var T:дерево);

{вспомогательная процедура: освободить память, занимаемую деревом T, и T:=nil}

begin

```

  if T<>nil then
    begin освоб(T^.лев); освоб(T^.прав);
          dispose(T); T:=nil
    end
end;

```

procedure упростить(var T: дерево);

var T1,T2: дерево;

begin

```

  if T^.элем in ['+', '-','*'] then
    begin

```

{сначала упростить операнды формулы:}

упростить(T^.лев); упростить(T^.прав);

T1:=T^.лев; T2:=T^.прав;

{упростить саму формулу:}

if (T^.элем='+') and (T1^.элем='0') or

(T^.элем='*') and (T1^.элем='1')

then {(0+f), (1*f) → f}

```
begin dispose(T1); dispose(T); T:=T2 end
else
if (T†.элем in ['+', '-']) and
(T2†.элем='0') or
(T†.элем='*') and (T2†.элем='1')
then {(f+0), (f-0), (f*1) → f}
begin dispose(T2); dispose(T); T:=T1 end
else
if (T†.элем='*') and
((T1†.элем='0') or (T2†.элем='0'))
then {(0*f), (f*0) → 0}
begin освоб(T†.лев); освоб(T†.прав);
T†.элем:='0' end
end
end
```

СПИСОК ЛИТЕРАТУРЫ

1. Абрамов В. Г., Трифонов Н. П., Трифонова Г. Н. Введение в язык Паскаль.— М.: Наука, 1988.— 320 с.
2. Абрамов С. А., Гнездилова Г. Г., Капустина Е. Н., Селюн М. И. Задачи по программированию.— М.: Наука, 1988.— 224 с.
3. Абрамов С. А., Зима Е. В. Начала программирования на языке Паскаль.— М.: Наука, 1987.— 112 с.
4. Агафонов В. Н., Поттосин И. В., Бежанова М. М., Сабельфельд В. К. Сборник упражнений по программированию на языке Паскаль.— Новосибирск: НГУ, 1985.— 80 с.
5. Бутомо И. Д., Самочадин А. В., Усанова Д. В. Программирование на алгоритмическом языке Паскаль для микроЭВМ.— Л.: ЛГУ, 1985.— 216 с.
6. Грагонюк П. Программирование на языке Паскаль.— М.: Мир, 1982.— 382 с.
7. Йенсен К., Вирт Н. Паскаль. Руководство для пользователя и описание языка.— М.: Финансы и статистика, 1982.— 150 с.
8. Касьянов В. Н., Сабельфельд В. К. Сборник заданий по практикуму на ЭВМ.— М.: Наука, 1986.— 272 с.
9. Любимский Э. З., Мартынюк В. В., Трифонов Н. П. Программирование.— М.: Наука, 1980.— 608 с.
10. Перминов О. Н. Программирование на языке Паскаль.— М.: Радио и связь, 1988.— 224 с.
11. Прайс Д. Программирование на языке Паскаль. Практическое руководство.— М.: Мир, 1987.— 232 с.
12. Трифонов Н. П. Сборник упражнений по Алголу.— М.: Наука, 1978.— 208 с.

Учебное издание

**Пильщиков Владимир Николаевич
СБОРНИК УПРАЖНЕНИЙ ПО ЯЗЫКУ ПАСКАЛЬ**

Заведующий редакцией А. С. Косов

Редактор О. И. Сухова

Художественный редактор Г. М. Коровина

Технический редактор С. Я. Шклад

Корректор И. Я. Кришталь

ИБ № 32692

Сдано в набор 27.10.88. Подписано к печати 09.08.89. Формат 84×108/32.
Бумага тип. № 2. Гарнитура литературная. Печать высокая. Усл. печ. л. 8,4.
Усл. кр.-отт. 8,82. Уч.-изд. л. 9,69. Тираж 260 000 экз. Заказ 9—406. Цена
35 коп.

**Ордена Трудового Красного Знамени издательство «Наука»
Главная редакция физико-математической литературы
117071 Москва В-71, Ленинский проспект, 15**

**Ордена Октябрьской Революции и ордена Трудового Красного Знамени МПО
«Первая Образцовая типография», Государственного комитета СССР по печати.
113054 Москва М-54, Валовая, 28**

**Отпечатано на полиграфкомбинате ЦК ЛКСМ Украины «Молодь» ордена Тру-
дового Красного Знамени издательско-полиграфическое объединение ЦК ВЛКСМ
«Молодая гвардия». 252119 Киев-119, ул. Пархоменко, 38—44.**

Vladimir N. Pilschikov, Cand. Sc. (Phys. & Math.)

EXERCISES IN PASCAL PROGRAMMING. Moscow, Nauka, Main Editorial Board for Physical and Mathematical Literature, 1989, 160 pp.

R e a d e r s h i p: students of mathematics and computer science.

C o n t e n t s: The book contains more than 900 exercises for those who begin learning the programming language Pascal and writing programs in this language. Many mathematical and data processing problems are proposed. The language version under consideration is the ISO Pascal standard.

A u t h o r: Lecturer on computer science at Moscow State University.

ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

ВЫЙДЕТ В СВЕТ В 1990 г.:

Дьяконов В. П. Форт-системы программирования персональных ЭВМ.

Является продолжением справочников автора, вышедших в издательстве «Наука». Впервые в отечественной литературе дается обстоятельное описание всех наиболее важных современных массовых версий языка Форт, включая расширенные версии, а также методики разработки на них пакетов прикладных программ. По широте охвата описанных версий Форт-систем программирования и их расширений, а также по числу конкретных примеров книга не имеет аналогов в мировой литературе.

Для научных и инженерно-технических работников, студентов вузов, программистов и массовых пользователей-непрофессионалов.

Заказы на книгу принимаются всеми магазинами Союзкниги и Академкниги, распространяющими физико-математическую литературу.

В Аннотированном тематическом плане выпуска литературы на 1990 г. книга В. П. Дьяконова занимает позицию 176.

ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

ВЫЙДЕТ В СВЕТ В 1990 г.:

Пярипу А. А. Программирование на алгоритмических языках.

Излагаются основы программирования на алгоритмических языках Бейсик, Фортран IV и ПЛ/1. Специальная глава содержит общие сведения о развитии ЭВМ, совершенствовании алгоритмических языков и принципах решения задач на ПЭВМ. Рассмотрены особенности входных языков некоторых широко используемых трансляторов.

2-е изд.— 1983 г.

Для студентов вузов.

Заказы на книгу принимаются всеми магазинами Союзкниги и Академкниги, распространяющими физико-математическую литературу.

В Аннотированном тематическом плане выпуска литературы на 1990 г. книга А. А. Пярипу занимает позицию 177.

ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

ВЫЙДЕТ В СВЕТ В 1990 г.:

Мамиконов А. Г., Кульба В. В., Косяченко С. А.,
Ужастов И. А. Оптимизация структур распределенных баз данных
в АСУ.

Изложены методы упорядочения и оптимизации структур распределенных баз данных (РБД) на этапах предпроектного анализа, технического проектирования и эксплуатации АСУ. Приведены модели, методы и алгоритмы анализа и структуризации предметных областей пользователей и синтеза оптимальных по различным критериям эффективности логических и физических структур РБД. Рассмотрены постановки задач, стратегии и методы реорганизации РБД.

Для разработчиков баз данных и специалистов, занимающихся разработкой автоматизированных систем проектирования и управления.

Заказы на книгу принимаются всеми магазинами Союзкниги и Академкниги, распространяющими физико-математическую литературу.

В Аннотированном тематическом плане выпуска литературы на 1990 г. книга А. Г. Мамиконова, В. В. Кульбы, С. А. Косяченко, И. А. Ужастова занимает позицию 166.

35 коп.

